

Πανεπιστήμιο Θεσσαλίας



Πολυτεχνική Σχολή

Τμήμα Μηχανικών Η/Υ, Τηλεπικοινωνιών και Δικτύων

Διπλωματική Εργασία

«Συγκριτική μελέτη και υλοποίηση κωδικοποιητών βίντεο με  
τα στάνταρ H.264 και VP8»

Δαμιανός Καλανίδης

Επιβλέπων καθηγητής

Ιωάννης Κατσαβουνίδης

Συνεπιβλέπων καθηγητής

Γεράσιμος Ποταμιάνος

Βόλος, Μάρτιος 2013



Ευχαριστίες,

Ευχαριστώ τον επιβλέποντα καθηγητή κ. Ιωάννη Κατσαβουνίδη για την πολύτιμη βοήθεια, καθοδήγηση και στήριξη του τόσο κατά την διάρκεια της διπλωματικής όσο και κατά την διάρκεια των υπόλοιπων μαθημάτων του, που αποτέλεσαν υπόβαθρο και πηγή έμπνευσης για την επιλογή του αντικείμενου μελέτης της διπλωματικής. Η παρουσία του στο τμήμα Τμήμα Μηχανικών Η/Υ, Τηλεπικοινωνιών και Δικτύων ως επιστήμονα, καθηγητή και ανθρώπου είναι πολύτιμη και αισθάνομαι τυχερός που τον είχα καθηγητή.

Ευχαριστώ τον συνεπιβλέποντα καθηγητή κ. Ποταμιάνο Γεράσιμο για τις συμβουλές, επισημάνσεις του κατά την διάρκεια της διπλωματικής.

Ευχαριστώ τον διδακτορικό φοιτητή κ. Πουλαράκη Στέργιο για την πολύτιμη και άμεση βοήθεια του κατά την διάρκεια της διπλωματικής. Οι συμβουλές, επισημάνσεις, γνώσεις του ήταν πολύτιμες.

Ευχαριστώ τους γονείς μου για την αμέριστη συμπαράσταση τους στην επιλογή μου να σπουδάσω το αντικείμενο που ήθελα από μικρή ηλικία.

Ευχαριστώ τους φίλους μου για τις συμβουλές τους, την βοήθεια τους και την ωραία παρέα που είχαμε κατά την διάρκεια των σπουδών μου.

Υγεία και ευτυχία σε όλο τον κόσμο.



## Περιεχόμενα

1	Εισαγωγή.....	7
2	Εισαγωγικές έννοιες.....	9
2.1	Αναπαράσταση εικόνας.....	9
2.2	Δειγματοληψία στο χώρο και στο χρόνο.....	9
2.3	Ψηφιακό βίντεο .....	11
2.4	Χρωματικά μοντέλα .....	11
2.5	Κωδικοποιητές και συμπίεση.....	12
2.5.1	Γενικές έννοιες .....	12
2.5.2	Κωδικοποιητές της οικογένειας Moving Picture Experts Group (MPEG).....	15
2.5.3	Κωδικοποιητές της On2 Technologies (On2 TrueMotion Video Compression) .....	17
2.5.4	Επιπρόσθετες έννοιες .....	18
2.5.5	Μέτρηση ποιότητας βίντεο.....	18
3	VP8.....	21
3.1	Τεχνική περιγραφή .....	23
3.1.1	Intra Frame Prediction.....	24
3.1.2	Motion Estimation .....	27
3.1.3	Inter Frame Prediction .....	28
3.1.4	Transformation .....	28
3.1.5	Quantization .....	29
3.1.6	Adaptive Loop Filtering .....	30
3.1.7	Entropy Coding .....	31
4	H.264 .....	33
4.1	Τεχνική περιγραφή .....	33
4.1.1	Τύποι frame .....	33
4.1.2	Intra Frame Prediction.....	35
4.1.3	Inter Frame Prediction.....	36
4.1.4	Quantization .....	39
4.1.5	Loop Filter.....	39
4.1.6	Entropy Coding .....	40
5	Android.....	43
6	Υλοποίηση και Αποτελέσματα δοκιμών .....	45
6.1	Υλοποίηση εκτελέσιμων VP8 σε Windows και Android.....	45
6.2	Υλοποίηση H.264.....	45

6.3	Υλοποίηση εργαλείων .....	45
6.4	Προδιαγραφές συστημάτων εκτέλεσης .....	45
6.5	Αποτελέσματα Μετρήσεων .....	47
6.5.1	Dataset properties .....	47
6.5.2	Αποδοτικότητα encoders .....	49
6.5.3	Αρχιτεκτονική 64 bit .....	69
6.5.4	Multithreading .....	70
7	Μελλοντικοί codec .....	74
7.1	H.265 .....	74
7.2	VP9 .....	80
8	Βιβλιογραφία .....	84

## 1 Εισαγωγή

Βίντεο είναι η προβολή συνεχόμενων εικόνων για την δημιουργία κίνησης σε συνδυασμό με ήχο. Τις τελευταίες δεκαετίες υπάρχει μια ραγδαία ανάπτυξη των πολυμέσων. Ειδικά τα τελευταία χρόνια με την έλευση συσκευών υψηλής ευκρίνειας, οι ανάγκες αποθήκευσης και υπολογιστικής ισχύος έχουν αυξηθεί. Συνεπώς απαιτούνται κατάλληλες μέθοδοι που θα ανταποκρίνονται στα νέα δεδομένα, με ιδιαίτερη έμφαση σε κωδικοποίηση που επιτυγχάνει μεγαλύτερο λόγο συμπίεσης.

Η εισαγωγή υλικού υψηλής ευκρίνειας στον μέσο χρήστη έγινε μαζικά με την υιοθέτηση της ψηφιακής τηλεόρασης υψηλής ευκρίνειας και κυρίως με την προώθηση των ψηφιακών δίσκων Blu-Ray. Το πρότυπο κωδικοποίησης που χρησιμοποιήθηκε ήταν ο κωδικοποιητής H.264, ο οποίος αποτελεί αντικατάσταση και μετεξέλιξη του παλαιότερου MPEG-2 που χρησιμοποιήθηκε στους ψηφιακούς δίσκους DVD. Πρόσφατα, έγινε ευρύτερα γνωστός ο κωδικοποιητής VP8, ο οποίος υπόσχεται να συναγωνιστεί τον H.264 σε επίπεδο ποιότητας με μεγαλύτερο πλεονέκτημα του ότι δεν έχει δικαιώματα χρήσης.

Στην παρούσα διπλωματική εργασία θα αναλυθούν οι προδιαγραφές και η απόδοση των δύο αυτών αλγορίθμων τόσο σε τυπικά συστήματα φορητών υπολογιστών αλλά και πως αποδίδουν όταν τροποποιούνται για χρήση σε φορητές συσκευές κάτι που είναι ιδιαίτερα ενδιαφέρον δεδομένου του προσανατολισμού καθιέρωσης του VP8 από τη Google στο διαδίκτυο.





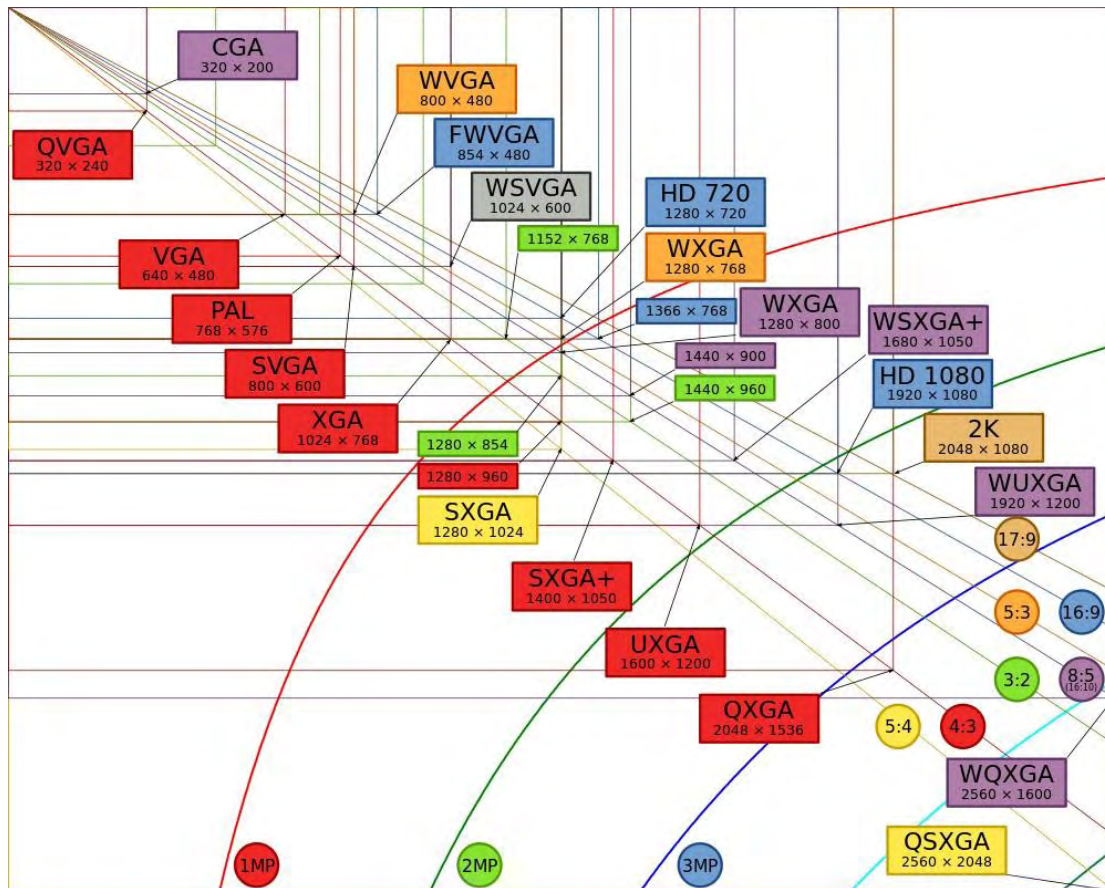
## 2 Εισαγωγικές έννοιες

### 2.1 Αναπαράσταση εικόνας

Βίντεο είναι η προβολή μιας τρισδιάστατης (3D) σκηνής σε ένα δισδιάστατο (2D) επίπεδο. Μια τρισδιάστατη (3D) σκηνή αποτελείται από ένα πλήθος αντικειμένων με βάθος, υφή και φωτισμό που προβάλλεται σε ένα επίπεδο για να δημιουργηθεί μια δισδιάστατη αναπαράσταση της σκηνής. Η δισδιάστατη αναπαράσταση περιέχει μεταβαλλόμενη υφή και φωτισμό αλλά καμιά πληροφορία βάθους. Μια ακίνητη εικόνα είναι ένα στιγμιότυπο της δισδιάστατης αναπαράστασης σε συγκεκριμένη στιγμή στο χρόνο. Μια ακολουθία εικόνων βίντεο αναπαριστά την σκηνή σε μια περίοδο χρόνου.

### 2.2 Δειγματοληψία στο χώρο και στο χρόνο

Μια πραγματική οπτική σκηνή είναι συνεχόμενη τόσο χρονικά όσο και χωρικά. Για να αναπαρασταθεί και να επεξεργαστεί μια οπτική σκηνή ψηφιακά, θα πρέπει να γίνει δειγματοληψία της πραγματικής οπτικής σκηνής τόσο χωρικά όσο και χρονικά. Χωρικά σε ένα ορθογώνιο πλέγμα στο επίπεδο της εικόνας βίντεο και χρονικά σαν μια σειρά από ακίνητες εικόνες (frames) δειγματοληπτούμενες ανά τακτά χρονικά διαστήματα. Οι διαστάσεις του χωρικού πλέγματος καθορίζουν την ανάλυση εικόνας. Μεγαλύτερη ανάλυση δίνει λεπτομερέστερη εικόνα αλλά δημιουργεί μεγαλύτερο όγκο δεδομένων. Στο παρακάτω σχήμα [Σχήμα 1] διακρίνονται τα κυριότερα μεγέθη αναλύσεων βίντεο.



Σχήμα 1: Γραφική αναπαράσταση των δημοφιλέστερων αναλύσεων εικόνας. Με μπλε χρώμα σημειώνονται αυτές με λόγο πλευρών 16:9 και με κόκκινο αυτές με λόγο πλευρών 4:3. Επίσης οι καμπύλες ορίζουν τα σημεία όπου η εικόνα έχει το αντίστοιχο μέγεθος σε MPixels

Η χρονική ανάλυση βασίζεται στο γεγονός ότι η αναπαραγωγή μιας σειράς από καρέ δίνει την ψευδαίσθηση της κίνησης, κυρίως λόγω της ιδιότητας του μεταισθήματος του ανθρώπινου οφθαλμού, δηλαδή ότι μια εικόνα παραμένει αποτυπωμένη στο μυαλό μας για περίπου 0,1sec αφού έχει σβήσει μπροστά από τα μάτια μας. Μονάδα μέτρησης του ρυθμού δειγματοληψίας των καρέ είναι το καρέ/δευτερόλεπτο (frames-per-second, fps). Όσο μεγαλύτερος είναι αυτός ο ρυθμός τόσο ομαλότερη είναι η κίνηση του βίντεο. Τυπικοί ρυθμοί αναπαραγωγής είναι τα 25 fps για την Ευρώπη, τα 30 fps για Αμερική και τα 24 fps για τις κινηματογραφικές ταινίες. Μια μέθοδος βελτίωσης της χρονικής οπτικής απεικόνισης χωρίς να αυξηθεί ο όγκος δεδομένων, αλλά εις βάρος της χωρικής ανάλυσης, είναι το πεπλεγμένο (interlaced) βίντεο ή αλλιώς και βίντεο εναλλασσόμενων γραμμών. Κάθε καρέ χωρίζεται σε δυο μέρη που το πρώτο αποτελείται από τις μονές γραμμές του βίντεο και το δεύτερο από τις ζυγές. Τα δυο αυτά μέρη ονομάζονται πεδία (fields). Πάντως αυτή η μέθοδος μπορεί να δημιουργήσει και οπτικά τεχνουργήματα, ειδικά αν η συσκευή αναπαραγωγής (οθόνη) δεν μπορεί να συνδυάσει την πληροφορία με τη σωστή χώρο-χρονική σειρά.

## 2.3 Ψηφιακό βίντεο

Ψηφιακό βίντεο είναι η αναπαράσταση μιας χρονοχωρικά δειγματοληπτούμενης σκηνής βίντεο σε ψηφιακή μορφή. Κάθε χωροχρονικό δείγμα, που ονομάζεται εικονοστοιχείο (pixel), αναπαριστάται ψηφιακά σαν ένας ή περισσότεροι αριθμοί οι οποίοι περιγράφουν την φωτεινότητα και τη χρωματικότητα του δείγματος.

Ένα ψηφιακό σύστημα βίντεο έχει ως εξής. Στην είσοδο του συστήματος, μια πραγματική οπτική σκηνή «συλλαμβάνεται», συνήθως από τον αισθητήρα μιας κάμερας και μετατρέπεται σε μια δειγματοληπτημένη ψηφιακή αναπαράσταση. Στην έξοδο του συστήματος το σήμα του ψηφιακού βίντεο προβάλλεται στον θεατή αναπαράγοντας την ακολουθία δισδιάστατων εικόνων βίντεο σε μια δισδιάστατη οθόνη, σαν μια χρονική ακολουθία.

## 2.4 Χρωματικά μοντέλα

Χρωματικό μοντέλο είναι ένα μαθηματικό μοντέλο που περιγράφει τον τρόπο που μπορούν τα χρώματα να αναπαρασταθούν ως σύνολο χρωματικών πλειάδων. Σε κάθε χρωματικό μοντέλο μπορούμε να εφαρμόσουμε μια συνάρτηση και να παραχθεί το ανάλογο χρωματικό διάστημα. Έστω ότι έχουμε ένα χρωματικό μοντέλο με 2 συνιστώσες  $X, Y$ . Μπορούμε να ορίσουμε ένα χρωματικό διάστημα του όπου τα  $X, Y$  παίρνουν τιμές από 1 έως 20 και ένα χρωματικό διάστημα του όπου τα  $X, Y$  παίρνουν τιμές από 18 έως 25. Οπότε μια πλειάδα με τιμές  $\langle 21, 24 \rangle$  δύναται να ανήκει στο 2<sup>ο</sup> χρωματικό διάστημα αλλά όχι στο 1<sup>ο</sup>.

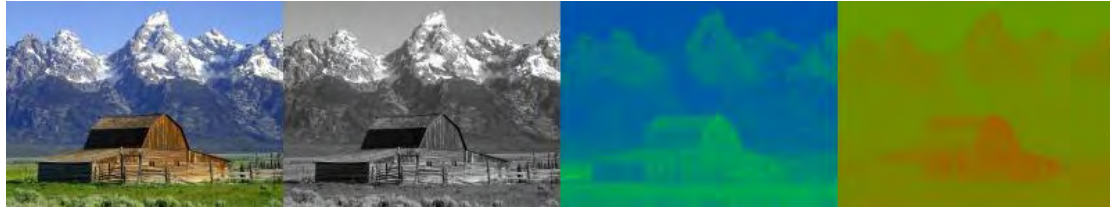
Οι πιο διαδεδομένοι τύποι χρωματικών μοντέλων είναι οι ακόλουθοι :

**RGB:** Είναι προσθετικό χρωματικό μοντέλο όπου τρία χρώματα, το κόκκινο, το πράσινο και το μπλε συνδυάζονται ώστε να αναπαράγουν ένα ευρύ φάσμα χρωμάτων. Είναι ευρέως διαδεδομένο στην προβολή και αναπαράσταση εικόνων σε ηλεκτρονικές συσκευές. Επίσης είναι device-dependent, δηλαδή κάθε συσκευή αναπαραγωγής εικόνων (οθόνη, προβολέας, εκτυπωτής) αποδίδει διαφορετικό χρωματικό αποτέλεσμα, ανάλογα με την τεχνολογία που χρησιμοποιεί (LCD, TFT, inkjet, laserjet).



Σχήμα 2: Αρχική εικόνα και ανάλυση της, στις R,G,B συνιστώσες της.

YCbCr: Παίρνει υπόψη του την ανθρώπινη αντίληψη για τα χρώματα. Αποτελείται από 3 συνιστώσες Y, Cb, Cr. Η Y είναι η συνιστώσα της φωτεινότητας και οι Cb, Cr είναι οι συνιστώσες της χρωματικότητας.



Σχήμα 3: Αρχική εικόνα και ανάλυση της, στις Y,Cb,Cr συνιστώσες της.

Το κύριο πλεονέκτημα του YCbCr σε σχέση με RGB είναι ότι οι Cb, Cr συνιστώσες έχουν μικρότερη ενέργεια στις υψηλές συχνότητες και γι' αυτό μπορούν να αναπαρασταθούν σε χαμηλότερη ανάλυση από τη συνιστώσα Y, ώστε να έχουμε λιγότερες απαιτήσεις δεδομένων για τις χρωματικές συνιστώσες χωρίς εμφανείς απώλειες στην οπτική ποιότητα.

Chroma subsampling: Είναι η μέθοδος κωδικοποίησης εικόνων που χρησιμοποιεί μικρότερη ανάλυση για την χρωματικότητα από ότι για την φωτεινότητα [3] και έτσι εκμεταλλευόμαστε το ανθρώπινο σύστημα όρασης που έχει μεγαλύτερη ευαισθησία στις διαφορές φωτεινότητας από ότι στις διαφορές χρωματικότητας.

## 2.5 Κωδικοποιητές και συμπίεση

### 2.5.1 Γενικές έννοιες

Κωδικοποιητής (encoder) είναι ένα μηχανήμα, κύκλωμα, συνάρτηση που μετατρέπει πληροφορία από ένα πεδίο σε κάποιο άλλο. Ο αποκωδικοποιητής (decoder) κάνει την αντίστροφη διαδικασία από τον κωδικοποιητή ώστε να επανέλθει η πληροφορία στην αρχική της μορφή. Ο αριθμός των bit που μεταφέρονται στην μονάδα του χρόνου ονομάζεται bitrate. Ως μονάδα μέτρησης ορίζεται το bits per second (bit/s ή bps). Στην επεξεργασία βίντεο χρησιμοποιούνται συνήθως πολλαπλάσια αυτής (Kbps ή Mbps). Λόγω του μεγάλου όγκου δεδομένων που απαιτεί το βίντεο συνήθως οι κωδικοποιητές χρησιμοποιούν συμπίεση ώστε να επιτύχουν ελάττωση του όγκου δεδομένων. Ο συνδυασμός κωδικοποιητή και αποκωδικοποιητή αποτελεί τον CODEC, (COmpressor-DECompressor)..

Είναι όμως απαραίτητη η συμπίεση δεδομένου των υψηλών χωρητικότητων των μέσων αποθήκευσης (σκληροί δίσκοι, flash drives) και μετάδοσης (DSL, VDSL); Θα δούμε ότι ένας αποδοτικός αλγόριθμος συμπίεσης έχει αρκετά πλεονεκτήματα τόσο σε χαμηλό όσο και υψηλό ρυθμό μετάδοσης δεδομένων, ανάλογα με το εύρος ζώνης του τηλεπικοινωνιακού καναλιού/δικτύου. Σε δίκτυα χαμηλού εύρους ζώνης επιτρέπει την χρήση εφαρμογών που αλλιώς θα ήταν αδύνατο να εκτελεστούν, για παράδειγμα σε μια τηλεφωνική γραμμή επιτρέπει βιντεοκλήσεις χαμηλής ανάλυσης. Σε κανάλια υψηλότερου εύρους ζώνης επιτρέπει την βελτίωση της εικόνας. Για παράδειγμα ένα DVD χωρητικότητας 4,7 GB μπορεί να αποθηκεύσει περίπου 2 ώρες ασυμπίεστου βίντεο ποιότητας QCIF (15fps) ή 2 ώρες συμπιεσμένου βίντεο ITU-R 601 (30fps). Προφανώς, οι πιο πολλοί χρήστες θα επέλεγαν το συμπιεσμένο βίντεο.

Διακρίνουμε δύο τύπους συμπίεσης, την απωλεστική και την μη απωλεστική. Στην μη-απωλεστική συμπίεση διατηρείται η ακεραιότητα των δεδομένων. Τα αρχικά δεδομένα μπορούν να ανακτηθούν επακριβώς μετά από συμπίεση και αποσυμπίεση. Στην απωλεστική τα αρχικά δεδομένα δεν μπορούν να ανακτηθούν πλήρως από τα συμπιεσμένα δεδομένα. Προφανώς για αρχεία κειμένου η απώλεια δεδομένων δεν είναι αποδεκτή. Ωστόσο, λόγω του ότι τα ανθρώπινα μάτια και τα αυτιά δεν μπορούν να διακρίνουν μικρές αλλαγές και λόγω του τεράστιου όγκου δεδομένων που απαιτεί το βίντεο, η απωλεστική συμπίεση σε αυτή την περίπτωση κρίνεται αποδεκτή και μερικές φορές αναγκαία λόγω τεχνικών και οικονομικών περιορισμών κατασκευής.

Στη συμπίεση βίντεο χρησιμοποιούνται κυρίως 3 τύποι καρτέ: I-frames, P-frames, και B-frames.

- Ένα I-frame είναι μια εσωτερικά κωδικοποιημένη εικόνα «Intra-coded picture» δηλαδή μια πλήρως και αυτόνομα ορισμένη εικόνα, όπως ένα συμβατικό στατικό αρχείο εικόνας, π.χ. τύπου JPEG.
- Ένα P-frame είναι μια προβλεπόμενη εικόνα «Predicted picture» που έχει μόνο τις αλλαγές της εικόνας σε σχέση με την εικόνα από προηγούμενο καρτέ. Για παράδειγμα, σε μια σκηνή όπου ένα αυτοκίνητο κινείται εμπρός από ένα ομοιόμορφο στατικό φόντο, μόνο η κίνηση του αυτοκινήτου θα κωδικοποιηθεί. Άρα ο κωδικοποιητής δεν χρειάζεται να αποθηκεύσει τα αμετάβλητα pixel του φόντου στο P-frame (για την ακρίβεια, κωδικοποιεί με ελάχιστα bits το ότι τα pixel φόντου έχουν μηδενική διαφορά από το προηγούμενο καρτέ).
- Ένα B-frame είναι μια διπλά προβλεπόμενη εικόνα «Bi-predictive picture» που εξοικονομεί ακόμη περισσότερο αποθηκευτικό χώρο και που χρησιμοποιούνται οι



διαφορές μεταξύ της τρέχουσας εικόνας και τόσο μεταγενέστερων όσο και προγενέστερων καρέ για να προσδιοριστεί το περιεχόμενό της.

Για να είναι πιο αποδοτική η συμπίεση βίντεο δεν κωδικοποιείται αυτούσιο το περιεχόμενο των προβλεπόμενων καρέ αλλά η διαφορά τους από τα εκάστοτε καρέ αναφοράς, ώστε να έχουμε ακόμα μεγαλύτερη εξοικονόμηση δεδομένων.

Θα μπορούσε κάποιος να ισχυριστεί ότι για να έχουμε τη μέγιστη συμπίεση αρκεί το πρώτο καρέ να είναι I-frame και τα άλλα P-frame. Αυτό είναι σωστό κατ' αρχάς, αλλά πρέπει κανείς να αναλογιστεί τι θα συμβεί αν κωδικοποιηθεί/αποσταλεί ή αποθηκευτεί εσφαλμένα το I-frame; Τότε όλα τα υπόλοιπα καρέ θα έχουν πρόβλημα και το πρόβλημα θα εντείνεται σε κάθε νέο καρέ. Το πρόβλημα είναι ακόμη μεγαλύτερο αν κανείς σκεφτεί εφαρμογές πραγματικού χρόνου, όπως η ψηφιακή τηλεόραση. Γι' αυτό σε μια ροή βίντεο έχουμε I-frame ανά τακτά χρονικά διαστήματα, θυσιάζοντας εξοικονόμηση χώρου προς όφελος της ανοχής σε λάθη και προσβασιμότητα σε τυχαίο χρόνο (random access). Φυσικά, το πόσο συχνά θα έχουμε I-frame εξαρτάται από διάφορους παράγοντες όπως το περιεχόμενο του βίντεο, το μέσο μετάδοσης/αναπαραγωγής κτλ.

Συνήθως οι εικόνες (καρέ) τμηματοποιούνται σε ορθογώνια/κομμάτια μικρότερου μεγέθους που ονομάζονται μάκρομπλοκ (macroblock) και μπορεί να επιλεγθεί σε επίπεδο μάκρομπλοκ ξεχωριστός τύπος πρόβλεψης χωρίς να χρειάζεται να επιλέξουμε ένα μόνο τύπο για όλη την εικόνα. Πιο συγκεκριμένα, τα I-frames περιέχουν μόνο intra μάκρομπλοκ, τα P-frames περιέχουν είτε intra μάκρομπλοκ είτε predicted μάκρομπλοκ και τα B-frames περιέχουν intra, predicted, ή bi-predicted μάκρομπλοκ.

Η πρόβλεψη με αντιστάθμιση κίνησης (motion compensation) περιγράφει μια εικόνα με βάση το μετασχηματισμό μιας εικόνας αναφοράς προς την τρέχουσα εικόνα. Η εικόνα αναφοράς μπορεί να προηγείται ή να έπεται της τρέχουσας εικόνας. Η αντιστάθμιση κίνησης εκμεταλλεύεται το γεγονός, που ισχύει ειδικά σε ταινίες, ότι είτε η κάμερα κινείται (pan) είτε ένα αντικείμενο κινείται στο κάδρο, με αποτέλεσμα να υπάρχει κοινή πληροφορία μεταξύ της εικόνας αναφοράς προς την τρέχουσα εικόνα, αρκεί κανείς να μετακινήσει κατάλληλα κομμάτια της εικόνας αναφοράς. Συνεπώς αν λάβουμε υπόψη μας στην πρόβλεψη του καρέ την αντιστάθμιση κίνησης, η διαφορά ειδικά σε επίπεδο macroblock μπορεί να είναι μηδενική ή σχεδόν μηδενική άρα και η πληροφορία προς συμπίεση και μετάδοση ελαχιστοποιείται αναλόγως.

## 2.5.2 Κωδικοποιητές της οικογένειας Moving Picture Experts Group (MPEG)

Το Moving Picture Experts Group (MPEG) είναι μια ομάδα εργασίας εμπειρογνομόνων που σχηματίστηκε από τον Διεθνή Οργανισμό Τυποποίησης (ISO) και τη Διεθνή Ηλεκτροτεχνική Επιτροπή (IEC) για τον καθορισμό προτύπων για τη συμπίεση και μετάδοση ήχου και βίντεο που ιδρύθηκε το 1988[3]. Ξεκίνησαν με την δημιουργία του MPEG-1 (1993) μέχρι τον πιο πρόσφατο H.264 (MPEG-4 AVC). Αναλυτικότερα :

### 2.5.2.1 MPEG-1 (1993)

Στόχος του συγκεκριμένου κωδικοποιητή απωλεστικής συμπίεσης ήταν η κωδικοποίηση κινούμενης εικόνας μαζί με ήχο για ψηφιακά μέσα αποθήκευσης για ροή δεδομένων έως και 1,5 Mbit/s σε ποιότητα παραπλήσια του διαδεδομένου – εκείνη την εποχή - αναλογικού φορμά (VHS) των βιντεοκασετών. Ήταν ο πρώτος υλοποιημένος codec του MPEG. Παρόλο που οι προδιαγραφές του επιτρέπουν να λειτουργήσει και για υψηλότερα bitrate, ο περιορισμός αυτός συνήθως επιδιωκόταν σκόπιμα ώστε να μπορεί να κωδικοποιηθεί και να αποθηκευτεί το βίντεο σύμφωνα με το διαθέσιμο bitrate των πρώτων οπτικών δίσκων (CD), οι οποίοι μέχρι τότε είχαν χρησιμοποιηθεί για αποθήκευση ασυμπίεστου ψηφιακού ήχου. Το πρότυπο που χρησιμοποιείται για αποθήκευση, εκτός από ήχο, και βίντεο κατά MPEG-1 σε οπτικούς δίσκους CD ονομάστηκε Video CD (VCD). Για την επίτευξη χαμηλού bitrate σε VCD γίνεται υποδειγματοληψία της αρχικής εικόνας του αναλογικού βίντεο από περίπου 352x480 (NTSC) ή 352x576 (PAL/SECAM) σε 352x240 (NTSC) ή 352x288 (PAL/SECAM). Ωστόσο η χρήση του δεν περιορίστηκε μόνο στα Video CD αλλά χρησιμοποιήθηκε για τις ψηφιακές εκπομπές καλωδιακής/δορυφορικής τηλεόρασης. Η μέγιστη υποστηριζόμενη ανάλυση του MPEG-1 είναι 4095×4095 (12-bits) και υποστηρίζει bitrates μέχρι και 100 Mbit/s. Χρησιμοποιεί το χρωματικό διάστημα YCbCr. Χρησιμοποιεί 3 τύπους συμπιεσμένων καρέ : I-frames, P-frames και τα D-frames. Τα D-frames είναι διαφορεικά μόνο (δηλαδή με μηδενικό διάνυσμα κίνησης) καρέ και συνέβαλαν στις μικρότερες ανάγκες επεξεργαστικής ισχύος για αυτήν την λειτουργία. Στις μεταγενέστερες υλοποιήσεις κωδικοποιητών αυτός ο τύπος frames καταργήθηκε. Επίσης χρησιμοποιεί μέγεθος 16x16 για τα macroblocks. Δημοσιεύτηκε το 1993 με τίτλο «Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s» (ISO/IEC 11172)

### 2.5.2.2 MPEG-2 (1995)

Αποτελεί μετεξέλιξη του MPEG-1. Χρησιμοποιήθηκε ευρέως στην ψηφιακή δορυφορική και καλωδιακή τηλεόραση αλλά και στους δίσκους DVD οι οποίοι έφεραν επανάσταση στην οικιακή ψυχαγωγία. Αποτελείται από 11 μέρη με το 2<sup>ο</sup> μέρος να είναι οι προδιαγραφές του για το βίντεο. Σε σχέση με το MPEG-1 πρόσφερε επιπλέον υποστήριξη για πεπλεγμένο βίντεο και υποστήριξη και για άλλα χρωματικά φορμά πέραν του 4:2:0. Αν και δεν ήταν βελτιστοποιημένος για βίντεο χαμηλού bitrate, σε bitrates μεγαλύτερα των 3 Mbit/s ήταν σαφώς ανώτερος του MPEG-1. Υποστηρίζει 3 τύπους συμπιεσμένων καρέ : I-frames, P-frames και B-frames, τα οποία αντικατέστησαν τα D-frames. Χρησιμοποιεί και αυτό μέγεθος 16x16 για τα macroblocks. Δεν ορίζει κάποια μέγιστη ανάλυση καρέ. Καθώς όμως υποστηρίζει μια πολύ μεγάλη γκάμα δυνατών εφαρμογών από φορητές συσκευές έως βίντεο υψηλής ανάλυσης και είναι μη-ρεαλιστικό και υψηλού κόστους να υλοποιηθεί από κάποια εφαρμογή το σύνολο των προδιαγραφών, το MPEG-2 ορίζει κάποιο υποσύνολο των δυνατοτήτων του και των τιμών των διαφόρων παραμέτρων ως «προφίλ» και «επίπεδα» προσαρμοσμένα για κάθε τύπο εφαρμογής. Έτσι καθένα από αυτά καθορίζει ένα ρεαλιστικό και αποδοτικό υποσύνολο του συνόλου των διαθέσιμων προδιαγραφών και κάθε εφαρμογή αναλόγως των απαιτήσεων της επιλέγει το κατάλληλο προφίλ και επίπεδο. Το προφίλ ορίζει το υποσύνολο των χαρακτηριστικών όπως το λόγο εμφάνισης καρέ (aspect ratio), το φορμά αναπαράστασης χρωματικότητας και το επίπεδο το υποσύνολο των ποσοτικών χαρακτηριστικών όπως μέγιστο ρυθμό μετάδοσης δεδομένων και ανάλυση. Το DVD κάνει συνδυασμό του Main Profile και Main Level (MP@ML) έχοντας όμως και επιπλέον περιορισμούς όπως για παράδειγμα στο bitrate που από 15 Mbps, το οποίο είναι το μέγιστο επιτρεπόμενο του Main Level, πέφτει στα 9.8 Mbps . Δημοσιεύτηκε το 1993 με τίτλο «Generic coding of moving pictures and associated audio information» (ISO/IEC 13818).

#### 2.5.2.3 MPEG-3 (Ενσωματώθηκε στο MPEG-2)

Ξεκίνησε ως εργασία πάνω στην υποστήριξη σημάτων 1080p υψηλού bitrate (20-40 Mbit/s) για την HDTV. Όμως σύντομα διαπιστώθηκε ότι μπορούσε να το κάνει αυτό και το ίδιο το MPEG-2, οπότε το 1992 το MPEG-3 ενσωματώθηκε σε αυτό. Αξίζει να σημειωθεί ότι το δημοφιλές στάνταρ συμπίεσης ήχου «MP3» ΔΕΝ αντιστοιχεί στο (ανύπαρκτο) στάνταρ MPEG-3, αλλά στο στάνταρ MPEG1-Layer III.

#### 2.5.2.4 MPEG-4 (1998)

Αποτελεί και αυτό με τη σειρά του μετεξέλιξη του MPEG-2 και χρησιμοποιεί περισσότερα εργαλεία κωδικοποίησης με αυξημένη πολυπλοκότητα για να επιτύχει μεγαλύτερο λόγο συμπίεσης. Επίσης περιλαμβάνει και διαχείριση πνευματικών δικαιωμάτων (DRM) που



ονομάζεται Intellectual Property Management and Protection (IPMP) και το MPEG-J, μια προγραμματιστική διεπαφή για την ανάπτυξη πολυμεσικών εφαρμογών βασιζόμενο στην γλώσσα java. Το τμήμα του που περιγράφει το βίντεο είναι τόσο το part 2 (“MPEG4-Video”) όσο και το part 10 και είναι ένα από αυτά που θα αναλυθούν στην παρούσα εργασία. Δημοσιεύτηκε το 1998 με τίτλο «Coding of audio-visual objects» (ISO/IEC 14496), όμως το εν λόγω part 10 δημοσιεύθηκε (παράλληλα με το ITU-T H.264) το 2003. Τέλος να σημειωθεί ότι για την εμπορική χρήση των κωδικοποιητών/αποκωδικοποιητών MPEG απαιτείται πληρωμή δικαιωμάτων στο MPEG Licensing Authority. Το part 2 του MPEG-4 δεν είναι ιδιαίτερα δημοφιλές και ανήκει στην ίδια οικογένεια με τα στάνταρ MPEG1/MPEG2, καθώς έχει πολλά κοινά χαρακτηριστικά, όπως η χρήση των ίδιων τύπων καρτέ (I-frame/P-frame/B-frame), το ίδιο μέγεθος μακρομπλόκ (16x16), τον ίδιο τύπο μετασχηματισμού (8x8 DCT) και παραπλήσιο τύπο διανυσμάτων κίνησης.

### 2.5.3 Κωδικοποιητές της On2 Technologies (On2 TrueMotion Video Compression)

Η On2 Technologies (πρώην “The Duck Corporation”) ιδρύθηκε το 1992. Ανέπτυξε μια σειρά κωδικοποιητών βίντεο με την ονομασία On2 TrueMotion Video Compression.

#### 2.5.3.1 1<sup>η</sup> γενιά On2 TrueMotion Video Compression

Ο πρώτος κωδικοποιητής, TrueMotion S (1993) στόχευε και χρησιμοποιήθηκε για βίντεο πλήρους κίνησης για παιχνίδια υπολογιστών. Ο TrueMotion RT (1996) προοριζόταν για επεξεργασία και σύλληψη βίντεο σε πραγματικό χρόνο. Ο TrueMotion 2 (1997) αποτέλεσε βελτιστοποίηση του TrueMotion S.

#### 2.5.3.2 2<sup>η</sup> γενιά On2 TrueMotion Video Compression

Τον Μάη του 2000 παρουσιάστηκε ο κωδικοποιητής βίντεο VP3.1 και μετά από τρεις μήνες ο κωδικοποιητής VP3.2. Το Σεπτέμβρη του 2001, η εταιρία On2 έκανε μια μεγάλη αλλαγή μετατρέποντας την εμπορική πολιτική της από κλειστό (δηλαδή διαθέσιμο επί πληρωμή) σε ανοικτό λογισμικό, διατεθώντας τον πηγαίο κώδικα και την άδεια χρήσης ανοικτού λογισμικού του VP3. Το VP4 (2001) αποτέλεσε μια βελτίωση του VP3. Μπορούσε να αναπαραχθεί αποκλειστικά από την εφαρμογή αναπαραγωγής βίντεο “RealPlayer” με συμφωνία παραχώρησης χρήσης που υπογράφηκε στις 19 Ιουλίου του 2001 και το Nullsoft Streaming Video φορμά της AOL. Αργότερα παρουσιάστηκε ο VP5 (2002). Τον Ιούνιο του

2002 η On2 δώρισε στο Xiph.org Foundation το VP3.2, υπό άδεια τύπου BSD που αποτέλεσε την γέννηση του κωδικοποιητή Theora. Ο VP6 (2003) και ο VP7 (2005) αποτέλεσαν τον βασικό κωδικοποιητή βίντεο του Macromedia Flash. Μάλιστα, ο VP6 υπήρξε υποψήφιος για το Enhanced Versatile Disc, τον ψηφιακό δίσκο βίντεο που ανέπτυξε η Κίνα ως εναλλακτικό του DVD. Ο VP7 επιλέχθηκε τον Δεκέμβριο του 2005 ως ένας από τους κωδικοποιητές βίντεο για βιντεοκλήσεις που χρησιμοποιεί η εφαρμογή κλήσεων Skype. Ο VP8 παρουσιάστηκε το 2008 και, μετά την εξαγορά της On2 από την γνωστή εταιρία Google το 2010, έγινε ανοικτού κώδικα με την υλοποίηση αναφοράς να δίδεται με άδεια BSD το 2010.

#### 2.5.4 Επιπρόσθετες έννοιες

**Intra Frame Prediction:** Είναι η μέθοδος όπου προσπαθούμε να προβλέψουμε το περιεχόμενο των μπλοκ στο τρέχον καρέ χωρίς να ανατρέξουμε σε άλλα καρέ. Χρησιμοποιεί ήδη κωδικοποιημένα block και pixels στο τρέχον καρέ.

**Διάνυσμα κίνησης (Motion vector)** είναι ένα δισδιάστατο διάνυσμα, που χρησιμοποιείται στο inter prediction, το οποίο παρέχει το offset των συντεταγμένων της αποκωδικοποιημένης εικόνας από τις συντεταγμένες της εικόνας αναφοράς.

**Motion Estimation** είναι η διαδικασία προσδιορισμού των διανυσμάτων κίνησης (motion vectors) που περιγράφουν τον μετασχηματισμό από μια 2D εικόνα σε μία άλλη.

**Motion Compensation:** Εφαρμογή των motion vectors στο προηγούμενο frame για την δημιουργία ενός «προβλέψιμου» καρέ. Αρκεί να μεταδοθεί η διαφορά μεταξύ «προβλέψιμου» και του καρέ αναφοράς.

**Inter Frame Prediction:** Χρησιμοποιώντας τα καρέ αναφοράς και τα motion vectors, προβλέπονται τα ενδιάμεσα (inter) καρέ.

**Quantization** είναι η διαδικασία συμπίεσης κατά την οποία, ένα εύρος τιμών αντιστοιχίζεται σε μια συγκεκριμένη τιμή.

#### 2.5.5 Μέτρηση ποιότητας βίντεο

Το πιο διαδεδομένο αντικειμενική μετρική ποιότητας είναι ο λόγος μέγιστου σήματος προς θόρυβο (PSNR). Μετριέται σε λογαριθμική κλίμακα και βασίζεται στο μέσο τετραγωνικό σφάλμα (MSE) μεταξύ της αυθεντικής και της επεξεργασμένης/συμπίεστης εικόνας.

$$PSNR = 10 \log_{10} \frac{(2^n - 1)^2}{MSE}$$

Όπου  $(2^n - 1)^2$  η μέγιστη δυνατή τιμή του pixel της εικόνας πχ Για εικόνα 8-bit ( $n=8$ ), η μέγιστη δυνατή τιμή του pixel είναι 255 και  $MSE = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$ . Στην προηγούμενη σχέση,  $I$  είναι η εικόνα αναφοράς και  $K$  η επεξεργασμένη – δηλαδή συμπίεσμένη και κατόπιν αποσυμπίεσμένη – εικόνα.

Στον παραπάνω υπολογισμό, χρησιμοποιούμε (συνήθως) τις τιμές φωτεινότητας του βίντεο. Επίσης, η ευκολία υπολογισμού του το έχει κάνει δημοφιλές μέτρο ποιότητας εικόνας του βίντεο. Υψηλό PSNR συνεπάγεται υψηλή ποιότητα βίντεο και χαμηλό συνεπάγεται χαμηλή ποιότητα βίντεο. Ωστόσο δεν συσχετίζεται καλά με υποκειμενικές μετρήσεις ποιότητας βίντεο όπως το ITU-R 500. Τυπικές τιμές PSNR που είναι καλό να έχει κανείς υπόψη του είναι ~30dB για «χαμηλής ποιότητας» βίντεο, ~40dB για «υψηλής (δηλαδή DVD) ποιότητας» βίντεο και ~50dB για «πολύ υψηλής», δηλαδή σχεδόν χωρίς απώλειες, βίντεο.



### 3 VP8

Το VP8 είναι μιας υψηλής απόδοσης τεχνολογία συμπίεσης βίντεο που αναπτύχθηκε από την εταιρία On2 Technologies. Η Google εξαγόρασε την On2 τον Φεβρουάριο του 2010. Το VP8 είναι ανοιχτού κώδικα και η υλοποίηση αναφοράς, που ονομάζεται libvpx, διατέθηκε το 2010 από την Google και έχει άδεια χρήσης τύπου BSD. Αναλυτικά η άδεια χρήσης έχει ως εξής :

## 1.1 Software License

Copyright (c) 2010, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

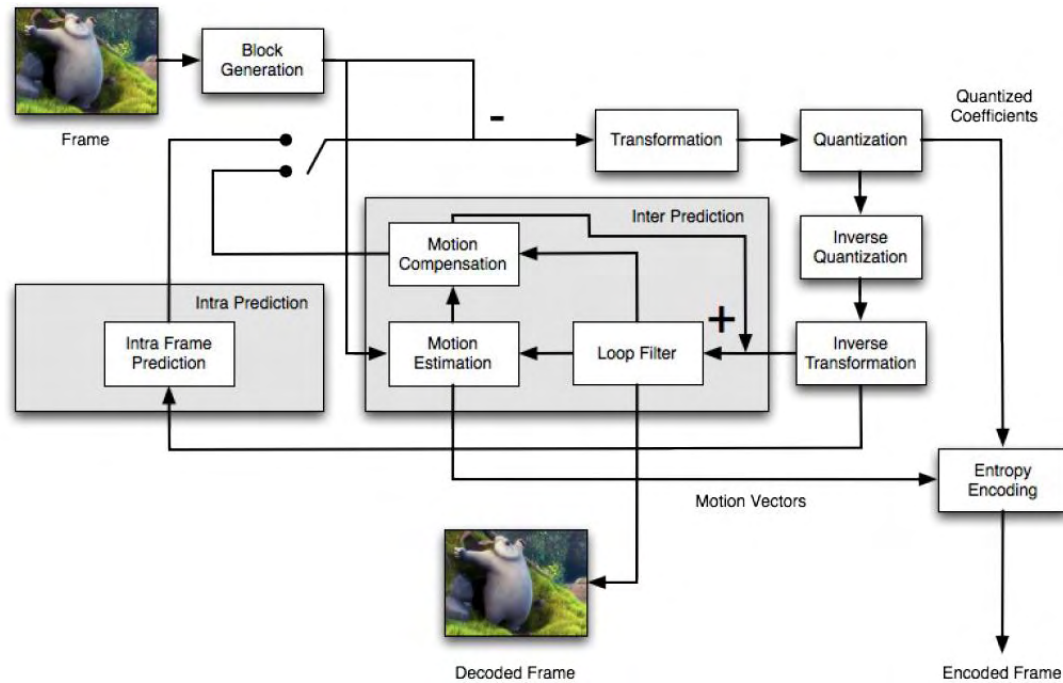
- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Google nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Η επιλογή αυτού του τύπου άδειας χρήσης από την Google δεν ήταν τυχαία καθώς τον κάνει ιδανικό για εφαρμογές διαδικτύου. Για αυτό η σχεδίαση του είναι βελτιστοποιημένη για τέτοιου είδους εφαρμογές.

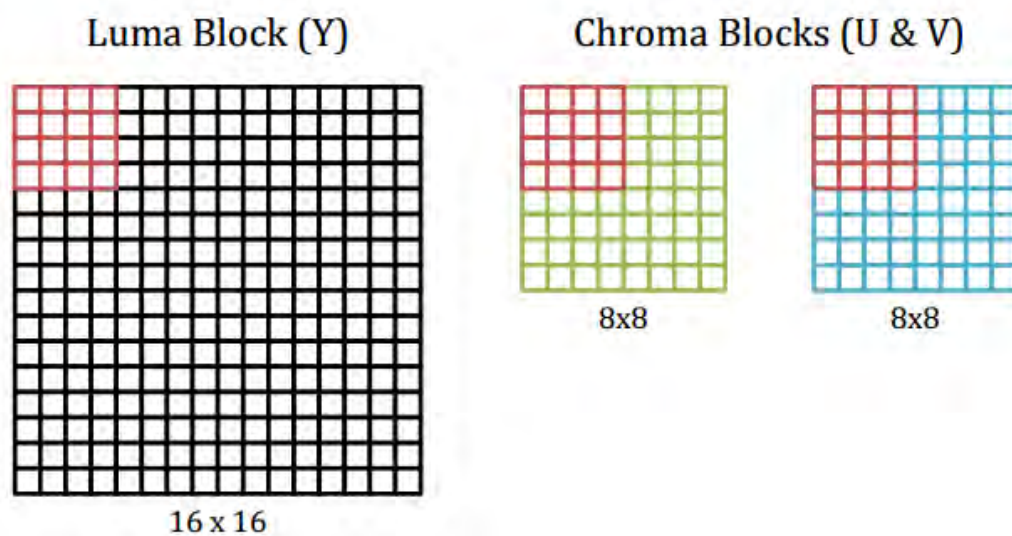
### 3.1 Τεχνική περιγραφή

Η δομή του VP8 απεικονίζεται στο παρακάτω διάγραμμα :



Σχήμα 4: Διάγραμμα ροής της πληροφορίας στον VP8 codec

Το VP8 χρησιμοποιεί για την συνιστώσα της φωτεινότητας (Y) μάκρομπλοκ μεγέθους 16x16 και για τις συνιστώσες της χρωματικότητας (U, V) μάκρομπλοκ μεγέθους 8x8.

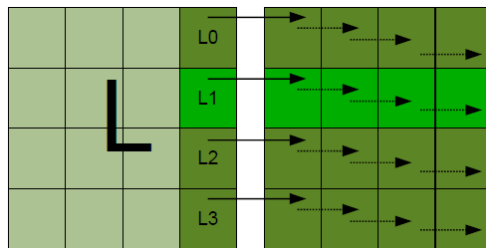


### 3.1.1 Intra Frame Prediction

Για τη φωτεινότητα (luma) έχουμε 4 επιλογές:

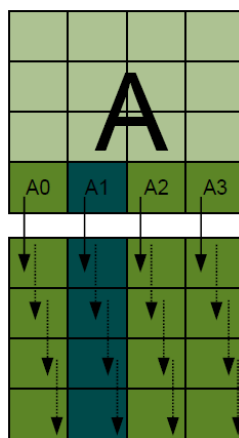
- H\_PRED

Συμπληρώνεται κάθε στήλη με ένα αντίγραφο της συνορεύουσας προς τα αριστερά στήλης.



- V\_PRED

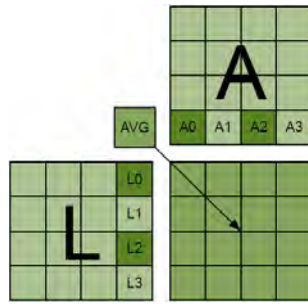
Συμπληρώνεται κάθε γραμμή με ένα αντίγραφο της συνορεύουσας προς τα πάνω γραμμής.



- DC\_PRED

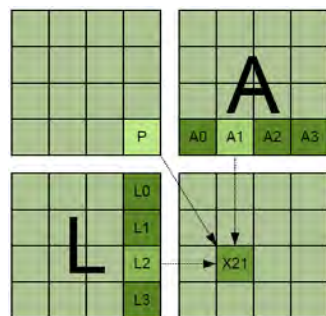
Χρησιμοποιείται ο μέσος όρος των pixel που βρίσκονται αριστερά και πάνω από το μπλοκ. Αν το μπλοκ βρίσκεται στην ανώτερη γραμμή του καρέ, χρησιμοποιείται ο μέσος όρος των αριστερών Pixel. Αν το μπλοκ βρίσκεται στην αριστερότερη στήλη του καρέ χρησιμοποιείται ο μέσος όρος των ανώτερων Pixel.





- TM\_PRED

$$X_{ij} = L_i + A_j - C \quad (i, j=0, 1, 2, 3)$$



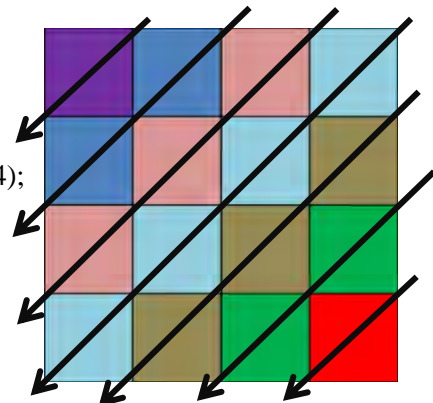
- B\_PRED

Κάθε 16x16 macroblock χωρίζεται σε (16) 4x4 macroblocks τα οποία προβλέπονται ανεξάρτητα, δηλαδή για καθένα από αυτά τα 4 μπορεί να χρησιμοποιηθεί οποιοδήποτε από τα διαθέσιμα B\_PRED modes πρόβλεψης.

Έχουμε 10 διαθέσιμα B\_PRED modes:

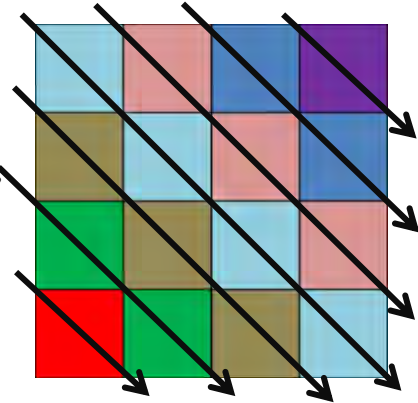
- B\_DC\_PRED: DC με χρήση των από πάνω και των αριστερών στηλών
- B\_TM\_PRED:  $X_{ij} = L_i + A_j - C \quad (i, j=0, 1, 2, 3)$
- B\_VE\_PRED: σειρές με αντίγραφο της συνορεύουσας πάνω σειράς
- B\_HE\_PRED: γραμμές με αντίγραφο της συνορεύουσας αριστερής γραμμής
- B\_LD\_PRED: southwest (left and down) 45 degree diagonal prediction

$$\begin{aligned} B[0][0] &= \text{avg3p}(A + 1); \\ B[0][1] &= B[1][0] = \text{avg3p}(A + 2); \\ B[0][2] &= B[1][1] = B[2][0] = \text{avg3p}(A + 3); \\ B[0][3] &= B[1][2] = B[2][1] = B[3][0] = \text{avg3p}(A + 4); \\ B[1][3] &= B[2][2] = B[3][1] = \text{avg3p}(A + 5); \\ B[2][3] &= B[3][2] = \text{avg3p}(A + 6); \\ B[3][3] &= \text{avg3}(A[6], A[7], A[7]); \end{aligned}$$



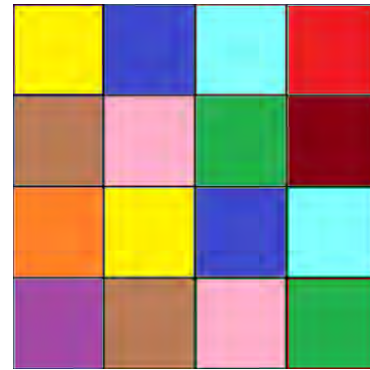
- B\_RD\_PRED: southeast (right and down)

$B[3][0] = \text{avg3p}(E + 1);$   
 $B[3][1] = B[2][0] = \text{avg3p}(E + 2);$   
 $B[3][2] = B[2][1] = B[1][0] = \text{avg3p}(E + 3);$   
 $B[3][3] = B[2][2] = B[1][1] = B[0][0] = \text{avg3p}(E + 4);$   
 $B[2][3] = B[1][2] = B[0][1] = \text{avg3p}(E + 5);$   
 $B[1][3] = B[0][2] = \text{avg3p}(E + 6);$   
 $B[0][3] = \text{avg3p}(E + 7);$



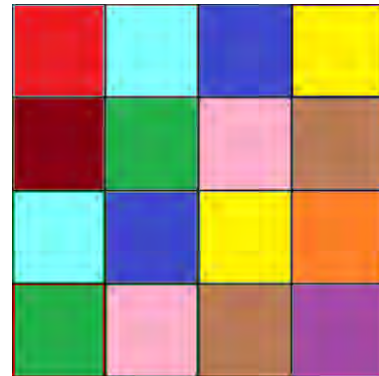
- B\_VR\_PRED: SSE (vertical right) diagonal

$B[3][0] = \text{avg3p}(E + 2);$   
 $B[2][0] = \text{avg3p}(E + 3);$   
 $B[3][1] = B[1][0] = \text{avg3p}(E + 4);$   
 $B[2][1] = B[0][0] = \text{avg2p}(E + 4);$   
 $B[3][2] = B[1][1] = \text{avg3p}(E + 5);$   
 $B[2][2] = B[0][1] = \text{avg2p}(E + 5);$   
 $B[3][3] = B[1][2] = \text{avg3p}(E + 6);$   
 $B[2][3] = B[0][2] = \text{avg2p}(E + 6);$   
 $B[1][3] = \text{avg3p}(E + 7);$   
 $B[0][3] = \text{avg2p}(E + 7);$



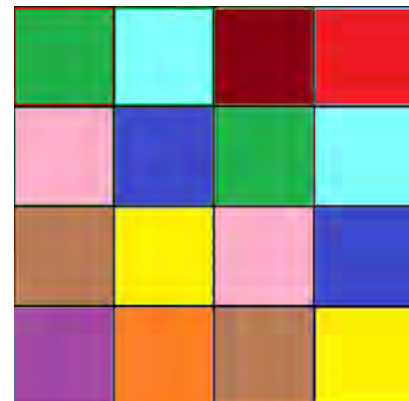
- B\_VL\_PRED: SSW (vertical left)

$B[0][0] = \text{avg2p}(A);$   
 $B[1][0] = \text{avg3p}(A + 1);$   
 $B[2][0] = B[0][1] = \text{avg2p}(A + 1);$   
 $B[1][1] = B[3][0] = \text{avg3p}(A + 2);$   
 $B[2][1] = B[0][2] = \text{avg2p}(A + 2);$   
 $B[3][1] = B[1][2] = \text{avg3p}(A + 3);$   
 $B[2][2] = B[0][3] = \text{avg2p}(A + 3);$   
 $B[3][2] = B[1][3] = \text{avg3p}(A + 4);$   
 $B[2][3] = \text{avg3p}(A + 5);$   
 $B[3][3] = \text{avg3p}(A + 6);$



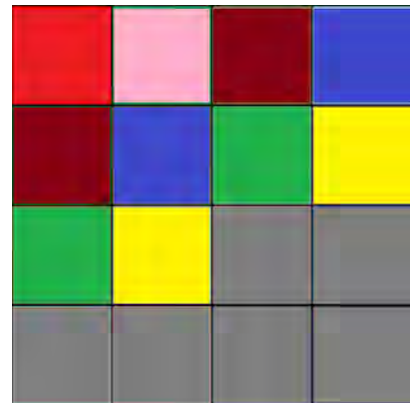
- B\_HD\_PRED: ESE (horizontal down)

$B[3][0] = \text{avg2p}(E);$  /\* predictor is from (5/2, -1) \*/  
 $B[3][1] = \text{avg3p}(E + 1);$  /\* (2, -1) \*/  
 $B[2][0] = B[3][2] = \text{avg2p}(E + 1);$  /\* (3/2, -1) \*/  
 $B[2][1] = B[3][3] = \text{avg3p}(E + 2);$  /\* (1, -1) \*/  
 $B[2][2] = B[1][0] = \text{avg2p}(E + 2);$  /\* (1/2, -1) \*/  
 $B[2][3] = B[1][1] = \text{avg3p}(E + 3);$  /\* (0, -1) \*/  
 $B[1][2] = B[0][0] = \text{avg2p}(E + 3);$  /\* (-1/2, -1) \*/  
 $B[1][3] = B[0][1] = \text{avg3p}(E + 4);$  /\* (-1, -1) \*/  
 $B[0][2] = \text{avg3p}(E + 5);$  /\* (-1, 0) \*/  
 $B[0][3] = \text{avg3p}(E + 6);$  /\* (-1, 1) \*/



- B\_HU\_PRED: ENE (horizontal up)

$B[0][0] = \text{avg2p}(L); /* \text{predictor is from } (1/2, -1) */$   
 $B[0][1] = \text{avg3p}(L + 1); /* (1, -1) */$   
 $B[0][2] = B[1][0] = \text{avg2p}(L + 1); /* (3/2, -1) */$   
 $B[0][3] = B[1][1] = \text{avg3p}(L + 2); /* (2, -1) */$   
 $B[1][2] = B[2][0] = \text{avg2p}(L + 2); /* (5/2, -1) */$   
 $B[1][3] = B[2][1] = \text{avg3}(L[2], L[3], L[3]); /* (3, -1) */$   
 $B[2][2] = B[2][3] = B[3][0] = B[3][1] = B[3][2] = B[3][3] = L[3]$



\*Όπου  $\text{avg3p}$  ο σταθμισμένος μέσος όρος τριών γειτονικών πίξελ κεντραρισμένων στο  $p$  :

Pixel  $\text{avg3p}(\text{const Pixel } *p) \{ \text{return avg3}(p[-1], p[0], p[1]); \}$

Για το chroma έχουμε διαθέσιμες τις 4 πρώτες επιλογές που είχαμε και για το luma :

- H\_PRED
- V\_PRED
- DC\_PRED
- TM\_PRED

### 3.1.2 Motion Estimation

Υπολογίζονται μέσω εκτίμησης κίνησης τα διανύσματα κίνησης μόνο των μπλοκ φωτεινότητας (Luma blocks). Τα διανύσματα κίνησης των μπλοκ χρωματικότητας (Chroma blocks) παράγονται από τα μπλοκ φωτεινότητας. Χρησιμοποιούνται διανύσματα κίνησης για μπλοκ μεγέθους 16x16, 16x8, 8x16, 8x8 και 4x4. Έχουμε ανάλυση  $\frac{1}{4}$  του pixel για φωτεινότητα και  $\frac{1}{8}$  του pixel για χρωματικότητα.

Χρησιμοποιούνται οι ακόλουθοι τύποι διανυσμάτων κίνησης (Motion Vectors) :

- MV\_NEAREST: Επαναχρησιμοποίηση μη-μηδενικού motion vector του τελευταίου αποκωδικοποιημένου μπλοκ.
- MV\_NEAR: Επαναχρησιμοποίηση μη-μηδενικού motion vector του προ-τελευταίου αποκωδικοποιημένου μπλοκ.
- MV\_ZERO: Το μπλοκ βρίσκεται στην ίδια θέση.

- MV\_NEW: Δημιουργία νέου motion vector.
- MV\_SPLIT: Το macroblock χωρίζεται σε υπο-μπλοκ με το κάθε ένα από αυτά να έχει το δικό του motion vector. Χρήσιμο για αντικείμενα εντός του macroblock με διαφορετικά χαρακτηριστικά κίνησης.

Στην συνέχεια έχουμε την λειτουργία του Motion Compensation, δηλαδή την εφαρμογή των motion vectors στο προηγούμενο frame για την δημιουργία ενός «προβλέψιμου» καρέ. Αρκεί να μεταδοθεί η διαφορά μεταξύ «προβλέψιμου» και πραγματικού καρέ.

### 3.1.3 Inter Frame Prediction

Χρησιμοποιώντας τα καρέ αναφοράς και τα motion vectors, προβλέπονται τα ενδιάμεσα (inter) καρέ. Σε αντίθεση με το H264, δεν έχουμε B frames.

#### Τρόπος ενδιάμεσης πρόβλεψης καρέ

Χρησιμοποιώντας κάποιο από τα ακόλουθα είδη καρέ:

- Previous Frame: Είναι το πιο πρόσφατο πλήρως αποκωδικοποιημένο καρέ.
- Alternate Reference Frame: Το εναλλακτικό αυτό frame μπορεί να είναι αόρατο. Μεταφέρει συμπληρωματική πληροφορία είτε για πρόβλεψη μείωσης θορύβου είτε για πρόβλεψη με κατεύθυνση στο παρελθόν η στο μέλλον. Στην 2<sup>η</sup> περίπτωση, σε συνδυασμό με τα Golden Frames, αντισταθμίζει την έλλειψη B-frames (το Alternate Reference Frame μεταδίδεται με πληροφορία από κάποιο μελλοντικό frame).
- Golden Reference Frame: Καρέ αναφοράς της εικόνας του τρέχοντος καρέ. Μπορεί να ενημερωθεί τμήμα του κάθε φορά. Χρήσιμο για ανάκτηση λαθών και για αλλαγή σκηνών.

### 3.1.4 Transformation

Χρησιμοποιείται υβριδικός μετασχηματισμός που συνδυάζει τον διακριτό μετασχηματισμό συνημιτόνου και το μετασχηματισμό Walsh-Hadamard..

### 3.1.5 Quantization

Η κβαντικοποίηση των συντελεστών του μετασχηματισμού DC είναι κλιμακωτή και βελτιστοποιημένη για ποιότητα μεταξύ 30dB to ~45dB SNR[7]. Έχουμε 6 διαφορετικούς πίνακες κβαντικοποίησης με 128 επίπεδα κβαντικοποίησης για κάθε πίνακα:

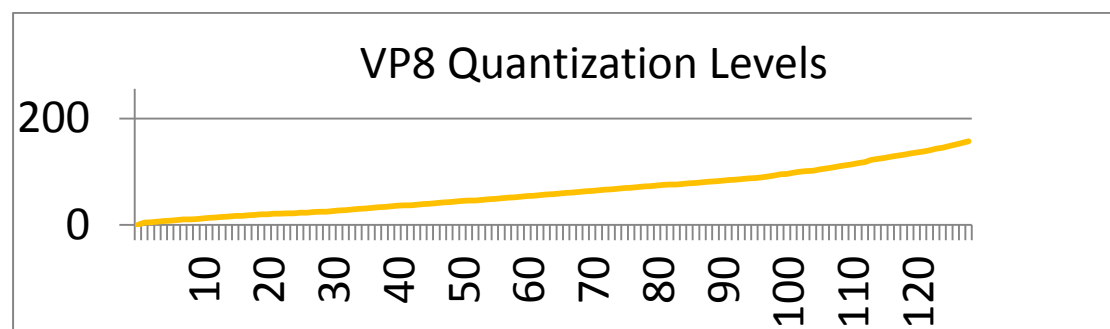
- ✓ 1st order luma DC
- ✓ 1st order luma AC
- ✓ 2nd order luma DC
- ✓ 2nd order luma AC
- ✓ Chroma DC
- ✓ Chroma AC

Ακολουθεί ο πίνακας κβαντικοποίησης 1st order luma DC στο σχήμα που ακολουθεί.

```
static const int dc_qlookup[QINDEX_RANGE] =
{
    4,  5,  6,  7,  8,  9, 10, 10, 11, 12, 13, 14, 15,
    16, 17, 17, 18, 19, 20, 20, 21, 21, 22, 22, 23, 23,
    24, 25, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
    36, 37, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 46,
    47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59,
    60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72,
    73, 74, 75, 76, 76, 77, 78, 79, 80, 81, 82, 83, 84,
    85, 86, 87, 88, 89, 91, 93, 95, 96, 98, 100, 101, 102,
    104, 106, 108, 110, 112, 114, 116, 118, 122, 124, 126, 128, 130,
    132, 134, 136, 138, 140, 143, 145, 148, 151, 154, 157
};
```

Σχήμα 5: Πίνακας κβαντικοποίησης 1st order luma DC VP8 128 επιπέδων

Η σχέση τιμών συντελεστών κβαντικοποίησης και επιπέδων κβαντικοποίησης είναι γραμμική όπως φαίνεται και στο παρακάτω σχήμα.



Σχήμα 6: Γράφημα σχέσης τιμών συντελεστών κβαντικοποίησης και επιπέδων κβαντικοποίησης

Για παράδειγμα για τον πίνακα συντελεστών  $A$  αν επιλέξουμε επίπεδο κβαντικοποίησης 3, ο αντίστοιχος συντελεστής κβαντικοποίησης είναι 6 και ο πίνακας  $Q$  που προκύπτει φαίνεται παρακάτω.

$$A = \begin{bmatrix} -312 & 7 & 1 & 0 \\ 1 & 12 & -5 & 2 \\ 2 & -3 & 3 & -1 \\ 1 & 0 & -2 & 1 \end{bmatrix} \quad Q = \text{round}\left(\frac{1}{6} * A\right) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 2 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Σχήμα 7: Παράδειγμα εφαρμογής τιμών συντελεστών κβαντικοποίησης

Οι κβαντοποιημένοι συντελεστές διαβάζονται με zig-zag σειρά. Επίσης χρησιμοποιείται Adaptive Quantization με συνέπεια το καρέ να μπορεί να χωριστεί σε 4 τμήματα το πολύ και για το καθένα εξ αυτών να εφαρμοσθεί διαφορετικό σκετ παραμέτρων κβαντικοποίησης.



Σχήμα 8: Εφαρμογή Adaptive Quantization

### 3.1.6 Adaptive Loop Filtering

Η μέθοδος του Adaptive Loop Filtering χρησιμοποιείται για την απαλοιφή των blocking artifacts (deblocking) που προκαλούνται από τις βασισμένες σε μπλοκ μεθόδους κωδικοποίησης του αλγορίθμου. Είναι ιδιαίτερα χρήσιμο όταν έχουμε υψηλή κβαντικοποίηση που δημιουργεί πολλά pixel με ίδιες τιμές. Είναι προσαρμοστικό (adaptive), δηλαδή εφαρμόζεται ανάλογα του επιπέδου κίνησης ξεχωριστά για κάθε περιοχή του καρέ. Επίσης είναι απαιτητικό σε υπολογιστική ισχύ (cpu-intensive) σε σχέση με τις υπόλοιπες μεθόδους του αποκωδικοποιητή κατά την αποκωδικοποίηση.

Στο VP8 έχουμε δύο ρυθμίσεις φίλτρου για το deblocking. Τη simple και τη normal.

Simple



- Εφαρμόζεται μόνο σε άκρα φωτεινότητας.
- Αγνοείται η παράμετρος `sharpness_level`
- Εφαρμόζεται όταν η διαφορά των μπλοκ ξεπερνά ένα όριο κατωφλίου
- 2 ή 3 συνεχόμενα pixel ανά μπλοκ

#### Normal

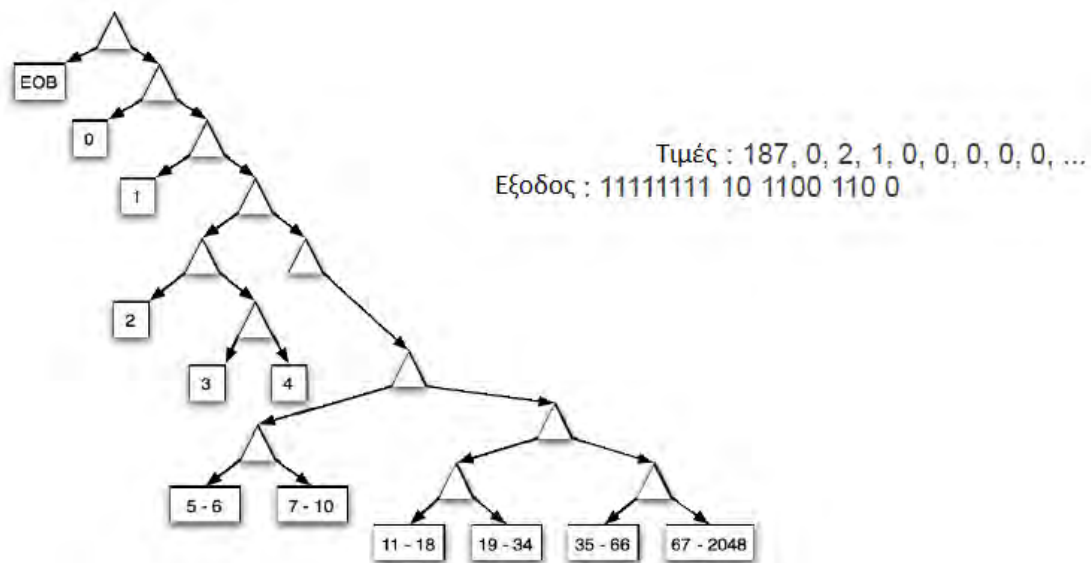
- 1,2,3 ή 4 συνεχόμενα pixel ανά γειτονικά μπλοκ

Στην υλοποίηση του codec υπάρχουν 2 σχετικές παράμετροι :

- `loop_filter_level` : Έχουμε 4 διαφορετικά επίπεδα για κάθε macroblock
- `sharpness_level` : Είναι ίδιο για όλο το καρέ

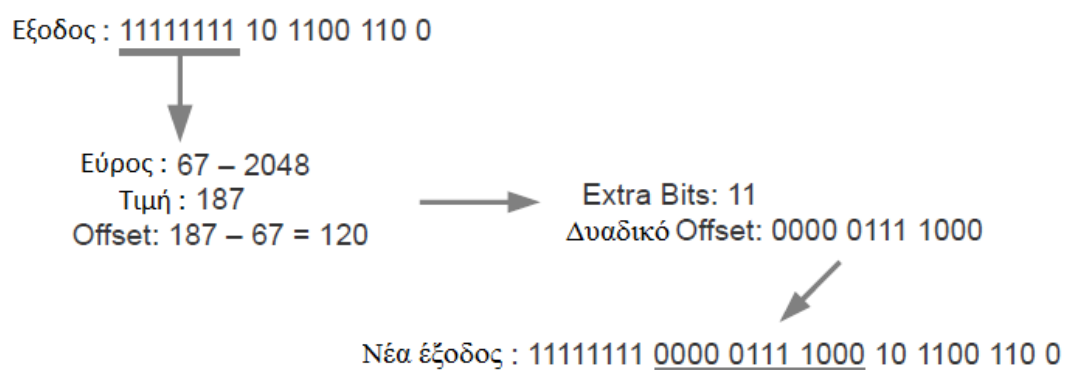
### 3.1.7 Entropy Coding

Με την εξαίρεση μερικών bit επικεφαλίδας που κωδικοποιούνται κατευθείαν σε ασυμπίεστη raw μορφή, η πλειοψηφία των συμπιεσμένων τιμών των δεδομένων κωδικοποιείται με την χρήση ενός αριθμητικού κωδικοποιητή Boolean. Ο αριθμητικού κωδικοποιητής Boolean κωδικοποιεί μία λογική τιμή Boolean (0/1) κάθε φορά. Χρησιμοποιείται για την μη απωλεστική κωδικοποίηση μιας ακολουθίας Boolean τιμών των οποίων η πιθανότητα να έχει τιμή 0 ή 1 μπορεί να εκτιμηθεί καλώς. Οι περισσότερες τιμές συμβόλων στον VP8 μετατρέπονται σε μια ακολουθία τιμών Boolean με την χρήση ενός σχήματος δένδρου ψευδός-Huffman. Σε αυτό το σχήμα, το δυαδικό δένδρο δημιουργείται πρώτα για ένα σύνολο συμβόλων παρόμοια με το πώς δημιουργείται ένα δένδρο Huffman και κάθε τιμή στο σύνολο μπορεί να αναπαρασταθεί με μια ακολουθία bit που παράγεται από την διαπέραση του δένδρου από την ρίζα προς τον αντίστοιχο κόμβο του φύλου.



Σχήμα 9: Σχήμα δένδρου ψευδό- Huffman

Επιπλέον, η διαφορά της τιμής με το εύρος του συνόλου δημιουργεί ένα offset το οποίο κωδικοποιείται με Binary arithmetic encoding.



Σχήμα 10: Τρόπος παραγωγής offset

Ο VP8 encoder υστερεί λόγω του carry propagation problem. Η υλοποίηση αναφοράς του H.264/AVC το επιλύει με τη χρήση OutstandingOneBit.. Η υλοποίηση αναφοράς VP8 το επιλύει με επεξεργασία των δεδομένων που έχουν ήδη γραφτεί στο bitstream προσθέτοντας 1 διαδοχικά μέχρι το κρατούμενο να επιλυθεί. Η μέθοδος OutstandingOneBit μπορεί επίσης να χρησιμοποιηθεί.



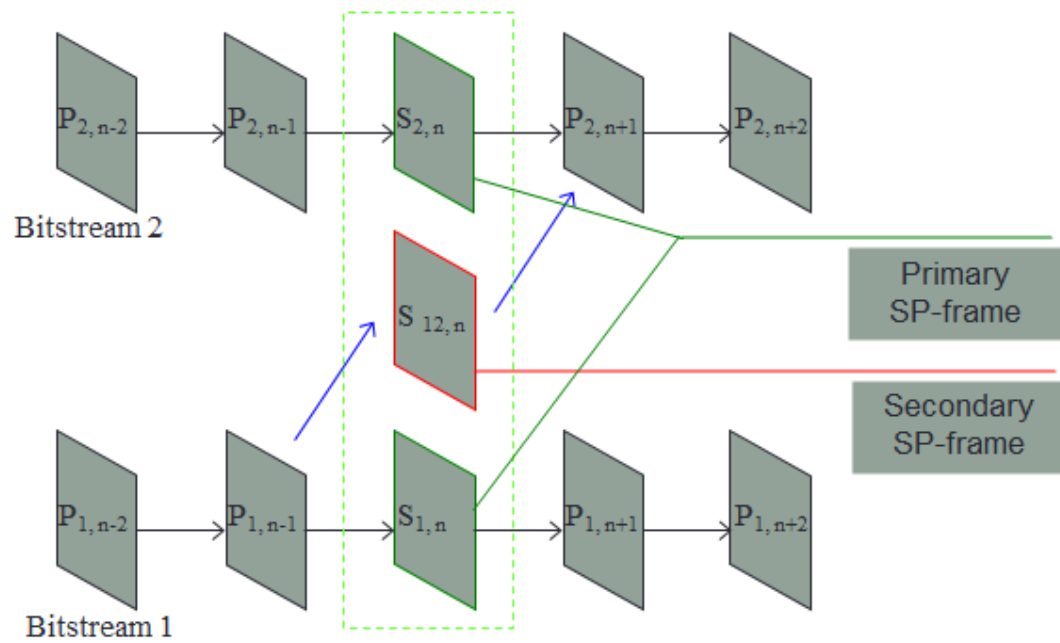
## 4 H.264

Το H.264 είναι το 10ο τμήμα (part 10) του στάνταρ MPEG-4 που περιγράφει το βίντεο. Είναι γνωστό και ως AVC (Advanced Video Coding). Έγινε ιδιαίτερα δημοφιλές με την χρήση του ως κωδικοποιητής στους ψηφιακούς δίσκους Blu-ray καθώς και στην ψηφιακή τηλεόραση (DVB-T, DVB-T2).

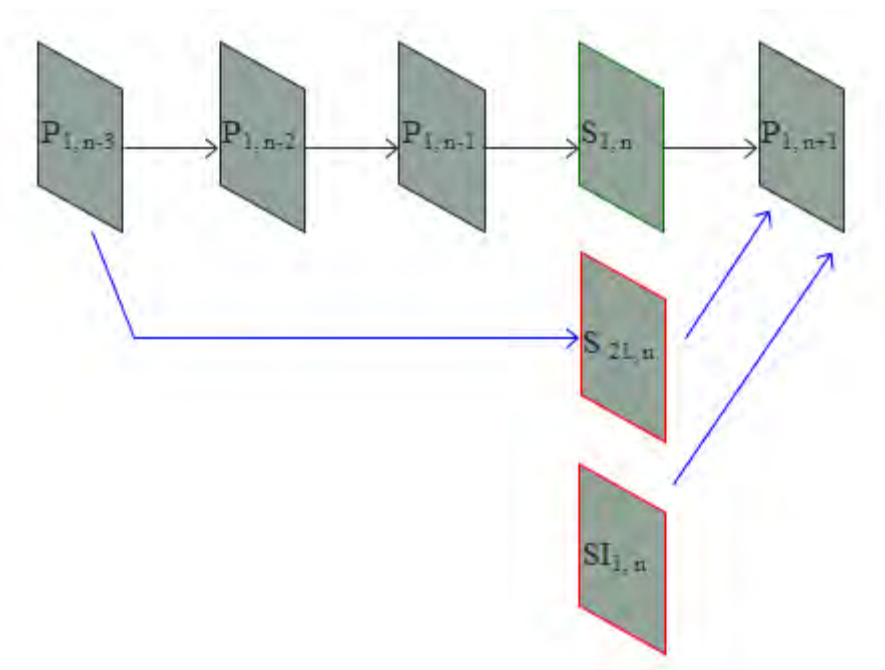
### 4.1 Τεχνική περιγραφή

#### 4.1.1 Τύποι frame

Έχουμε τους εξής τύπους frames : I-frames, P-frames, και B-frames και επιπλέον το H.264 ορίζει δύο νέους τύπους καρέ που αναφέρονται ως SP και SI frames στο Extended Profile. Τα SP frames χρησιμοποιούν πρόβλεψη με αντιστάθμιση κίνησης για να εκμεταλλευτούν το χρονικό πλεονασμό στην ακολουθία όπως και τα P-frames και επιπλέον επιτρέπουν πανομοιότυπα frames να ανακατασκευάζονται ακόμα και αν προβλέπονται από διαφορετικά καρέ αναφοράς. Επίσης απαιτούν λιγότερα bits από τα I-frames. Τα SI frames χρησιμοποιούν χωρική (spatial) πρόβλεψη όπως και τα I-frames και μπορούν να συνδυαστούν με τα SP frames. Τα SP-frames και SI-frames μπορούν να χρησιμοποιηθούν αντί για τα I-frames σε εφαρμογές όπως εναλλαγή bitstream (bitstream switching), ένωση (splicing), τυχαία πρόσβαση (random access) , προεπισκόπηση προς τα μπρος ή προς τα πίσω (fast forward/fast backward) και ανθεκτικότητα/ανάκτηση λαθών (error resilience/recovery).



Σχήμα 11: Παράδειγμα εναλλαγής bitstream με SP-frames



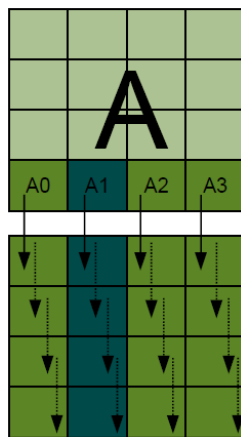
Σχήμα 12: Παράδειγμα χρήσης SI-frames για Error Recovery/Resiliency

#### 4.1.2 Intra Frame Prediction

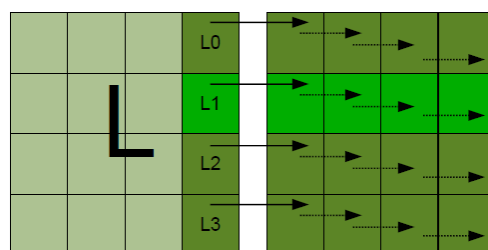
Είναι η μέθοδος όπου προσπαθούμε να προβλέψουμε το περιεχόμενο των μπλοκ στο τρέχον καρέ χωρίς να ανατρέξουμε σε άλλα καρέ. Χρησιμοποιεί ήδη κωδικοποιημένα block στο τρέχον καρέ. Έχουμε 4 τρόπους ανάλυσης (modes) για 16x16 μπλοκ φωτεινότητας, 9 τρόπους για 4x4 μπλοκ φωτεινότητας και 4 τρόπους για τα μπλοκ χρωματικότητας. Το στάνταρ H.264 μοιάζει σημαντικά με το VP8, στο κομμάτι αυτό, συγκεκριμένα, το H.264 προσφέρει μεγαλύτερη ποικιλία τρόπων κωδικοποίησης Intra Frame Prediction από το VP8.

##### 16x16 μπλοκ φωτεινότητας

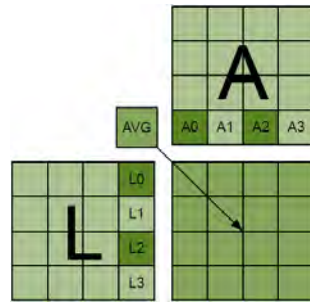
- Mode 0 (vertical)



- Mode 1 (horizontal)



- Mode 2 (DC)



- Mode 3 (Plane).

#### 4x4 μπλοκ φωτεινότητας

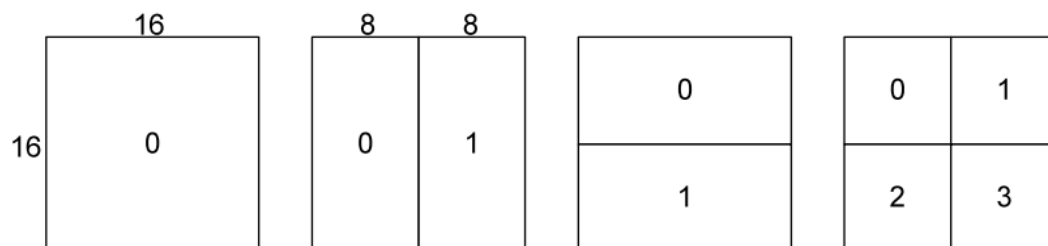
0 (vertical), 1 (horizontal), 2 (DC), 3 (diag down/left), 4 (diag down/right), 5 (vertical/right), 6 (horizontal/down), 7 (vertical/left), 8 (horizontal/up)

#### 8x8 μπλοκ χρωματικότητας

Όπως και πιο πάνω Mode 0 (vertical), Mode 1 (horizontal), Mode 2 (DC), Mode 4 (Plane).

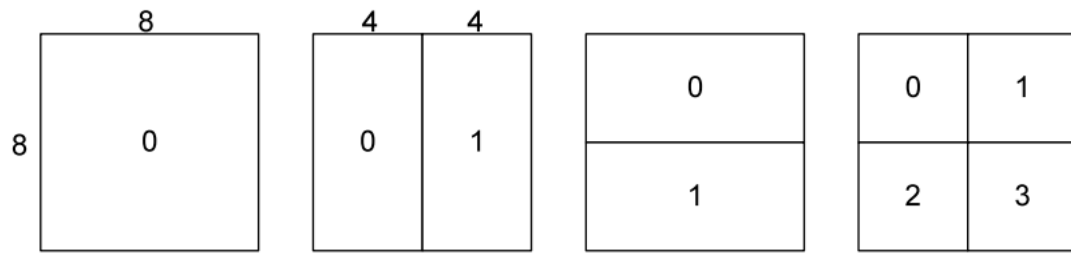
### 4.1.3 Inter Frame Prediction

Υποστηρίζονται μπλοκ φωτεινότητας μεγέθους από 16x16 μέχρι 4x4 με διάφορες επιλογές μεταξύ των 2. Έχουμε για κάθε μακροβλοκ 16x16 τις βασικές κατατμήσεις 16x16, 8x16, 16x8, 8x8 με κάθε υποπεριοχή να ονομάζεται κατάτμηση μάκρομπλοκ (macroblock partition)



Σχήμα 13: Βασικές κατατμήσεις μακροβλοκ 16x16

και τις υπο-κατατμήσεις 8x8, 4x8, 8x4, 4x4 αν επιλεγεί ως βασική κατάτμηση η 8x8.

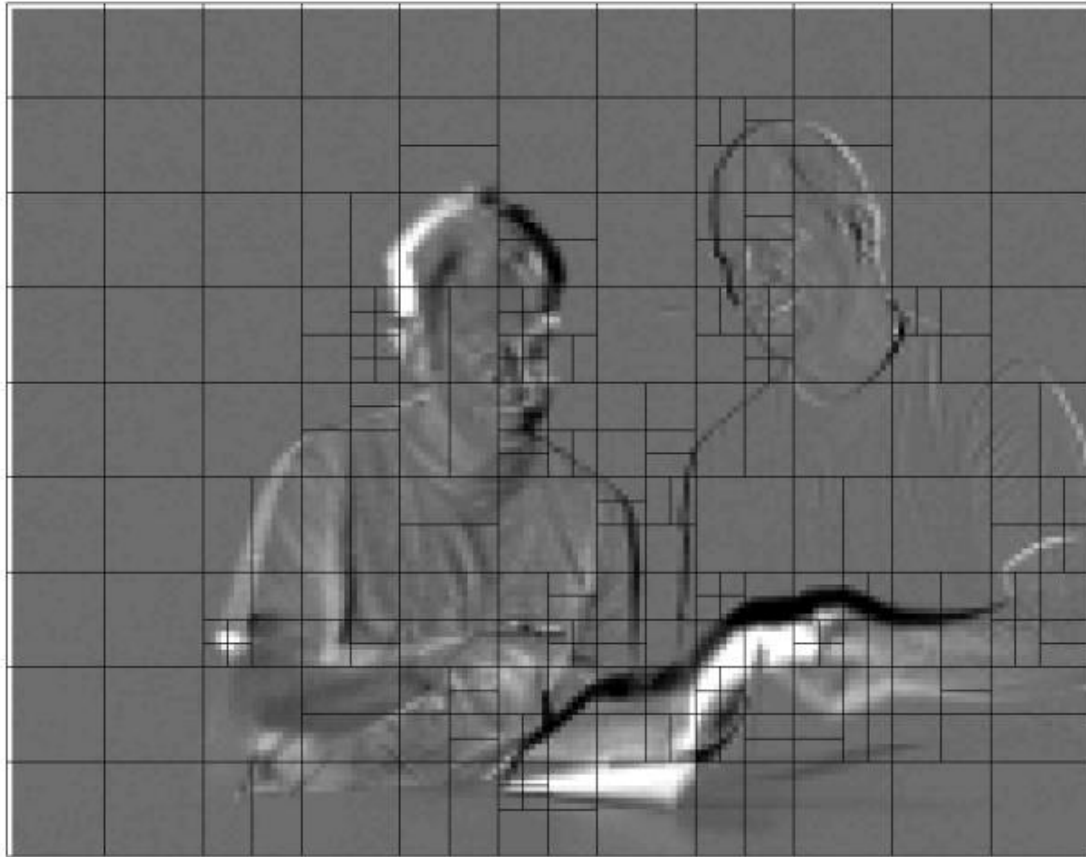


Σχήμα 14: Υπο-κατατμήσεις για βασική κατάτμηση 8x8.

Αυτή η μέθοδος κατάτμησης μακρομπλοκ σε motion estimated - υπο-μπλοκ - μεταβλητού μεγέθους είναι γνωστή και ως tree structured motion compensation.

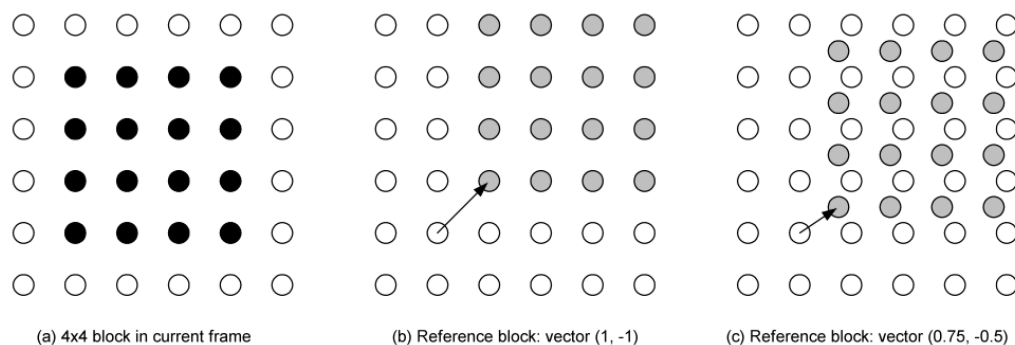
Η ανάλυση για κάθε συνιστώσα χρωματικότητας σε ένα μακρομπλοκ είναι η μισή της συνιστώσας φωτεινότητας. Κάθε μπλοκ χρωματικότητας έχει κατατμήσεις όπως το μπλοκ φωτεινότητας με την διαφορά ότι κάθε μια από αυτές έχουν την μισή ανάλυση οριζόντια και κάθετα. Για παράδειγμα μια κατάτμηση χρωματικότητας 4x8 αντιστοιχεί σε κατάτμηση φωτεινότητας 8x16. Επίσης η οριζόντια και κάθετη συνιστώσα κάθε διανύσματος κίνησης μειώνεται στο μισό όταν εφαρμόζεται σε μπλοκ χρωματικότητας.

Για κάθε κατάτμηση ή υπομπλοκ απαιτείται ξεχωριστό διάνυσμα κίνησης (motion-vector). Επιλέγοντας ένα μεγάλο μέγεθος κατάτμησης συνεπάγεται λιγότερος αριθμός bits για την αποστολή των διανυσμάτων κίνησης και του τύπου των κατατμήσεων. Ωστόσο το περίσσευμα από την αντιστάθμιση κίνησης ίσως να περιέχει σημαντικό ποσό ενέργειας σε περιοχές του καρέ με υψηλή λεπτομέρεια. Αντιθέτως, επιλέγοντας μικρό μέγεθος κατάτμησης συνεπάγεται περισσότερα bits για την αποστολή των διανυσμάτων κίνησης και του τύπου των κατατμήσεων πράγμα που ίσως είναι περιττό σε περιοχές του καρέ με χαμηλή λεπτομέρεια όπου το περίσσευμα από την αντιστάθμιση κίνησης ίσως να περιέχει αμελητέο ποσό ενέργειας. Μια βέλτιστη επιλογή κατατμήσεων φαίνεται στην παρακάτω εικόνα όπου οι πιο σκούρες περιοχές έχουν εντονότερη κίνηση.



Σχήμα 15: Διαχωρισμός εικόνας σε μπλοκ.

Κάθε κατάτμηση σε ένα inter-coded μακρομπλοκ προβλέπεται από μια ισομεγέθη περιοχή στην εικόνα αναφοράς. Η απόκλιση (motion vector) μεταξύ των δυο περιοχών έχει ανάλυση  $\frac{1}{4}$ -pixel για την συνιστώσα φωτεινότητας και ανάλυση  $\frac{1}{8}$ -pixel για τις δύο συνιστώσες χρωματικότητας. Αν η οριζόντια και κάθετη συνιστώσα του διανύσματος κίνησης είναι ακέραιοι αριθμοί, τότε τα σχετικά δείγματα στο καρέ αναφοράς είναι υπαρκτά pixel. Αν όμως μια ή και οι δυο συνιστώσες είναι δεκαδικοί αριθμοί, τότε τα σχετικά δείγματα προβλέπονται - μέσω φίλτρου 6-tap separable για pixel φωτεινότητας ή με δι-γραμμική παρεμβολή για pixel χρωματικότητας - μεταξύ των παρακείμενων pixel στο καρέ αναφοράς.

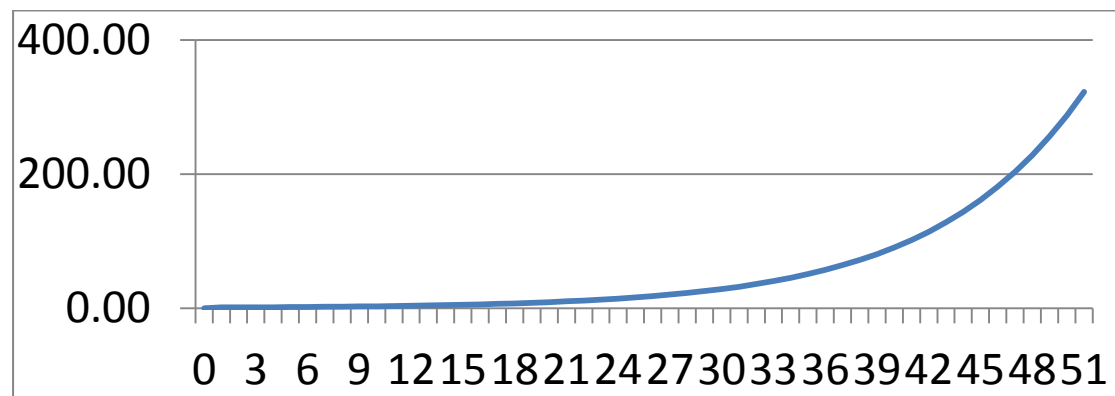


Σχήμα 16: Πρόβλεψη για κλασματική απόκλιση pixels

Η αντιστάθμιση κίνησης υπό-πίξελ δίνει εμφανώς καλύτερη αποδοτικότητα συμπίεσης από την αντιστάθμιση κίνησης ακέραιου-πίxel με το κόστος όμως της πολυπλοκότητας υπολογισμού.

#### 4.1.4 Quantization

Η κβαντικοποίηση των συντελεστών του DC μετασχηματισμού είναι εκθετική και έχουμε 52 επίπεδα κβαντικοποίησης.



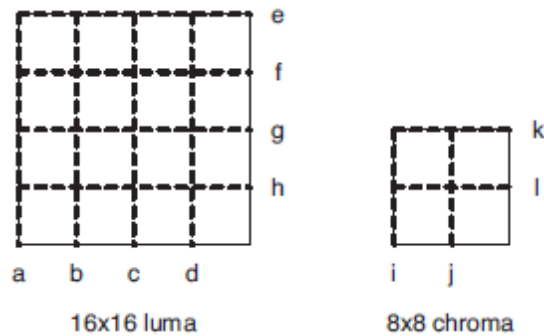
Σχήμα 17: Γράφημα σχέσης τιμών συντελεστών κβαντικοποίησης και επιπέδων κβαντικοποίησης

#### 4.1.5 Loop Filter

Εφαρμόζεται σε κάθε μάκρομπλοκ για να ελαττώσει την παραμόρφωση λόγω της επεξεργασίας του καρέ ανά μπλοκ. Το φίλτρο εφαρμόζεται μετά τον αντίστροφο μετασχηματισμό στον κωδικοποιητή και πριν την επανασύνθεση και αποθήκευση του μάκρομπλοκ για μελλοντικές προβλέψεις και στον αποκωδικοποιητή πριν την επανασύνθεση και προβολή του μάκρομπλοκ. Το φίλτρο λειαίνει τα άκρα του μάκρομπλοκ βελτιώνοντας την οπτική απεικόνιση των αποκωδικοποιημένων καρέ. Η φιλτραρισμένη εικόνα χρησιμοποιείται κατά την αντιστάθμιση κίνησης μελλοντικών καρέ, κάτι που επιδρά θετικά στην απόδοση συμπίεσης επειδή η φιλτραρισμένη εικόνα είναι πιο πιστή αναπαραγωγή του αρχικού καρέ.

Το φίλτρο εφαρμόζεται σε οριζόντια η κατακόρυφα άκρα των 4×4 μπλοκ σε ένα μάκρομπλοκ εκτός από τα άκρα στα όρια του καρέ, με την ακόλουθη σειρά :

1. Φιλτράρισμα 4 κάθετων ορίων της συνιστώσας φωτεινότητας με τη σειρά a,b,c,d
2. Φιλτράρισμα 4 οριζοντίων ορίων της συνιστώσας φωτεινότητας με τη σειρά e,f,g,h
3. Φιλτράρισμα 2 κάθετων ορίων κάθε συνιστώσας χρωματικότητας (i,j)
4. Φιλτράρισμα 2 οριζοντίων ορίων κάθε συνιστώσας χρωματικότητας (k,l)



Σχήμα 17: Εφαρμογή Loop Filter

Κάθε διαδικασία φιλτραρίσματος επηρεάζει μέχρι 3 pixel το πολύ σε κάθε πλευρά του ορίου και η ένταση του φίλτρου εξαρτάται από την τρέχουσα παράμετρο QP του κβαντοποιητή, τους τύπους κωδικοποίησης των γειτονικών μπλοκ και την κλίση των δειγμάτων εικόνας κατά μήκος του ορίου. Η ισχύς του φίλτρου είναι μεγαλύτερη σε σημεία όπου είναι πιθανότερο να έχουμε μεγαλύτερη παραμόρφωση λόγω blocking.

#### 4.1.6 Entropy Coding

Το πρότυπο H.264/AVC προσδιορίζει αρκετές διαφορετικές μεθόδους για την κωδικοποίηση συμβόλων ώστε να μεταφερθούν ή να αποθηκευτούν στη ροή δεδομένων. Αυτές οι μέθοδοι είναι οι ακόλουθες:

- **Fixed length code:** Ένα σύμβολο μετατρέπεται σε ένα δυαδικό κώδικα προσδιορισμένου μήκους (n bit).
- **Exponential-Golomb variable length code:** Το σύμβολο αναπαρίσταται σαν μια κωδική λέξη Exp-Golomb μεταβλητού αριθμού bits (v bits). Γενικότερα, μικρότερου μήκους κωδικές λέξεις Exp-Golomb ανατίθενται σε σύμβολα που έχουν μεγαλύτερη συχνότητα εμφάνισης.
- **CAVLC:** Context-Adaptive Variable Length Coding, μια ειδικά σχεδιασμένη μέθοδος κωδικοποίησης συντελεστών μετασχηματισμού όπου διαφορετικά σετ



κωδικών μεταβλητού μήκους επιλέγονται ανάλογα με τα στατιστικά των πρόσφατα κωδικοποιημένων συντελεστών, με χρήση προσαρμογής συμφραζομένων.

- **CABAC:** Context-Adaptive Binary Arithmetic Coding, μια μέθοδος αριθμητικής κωδικοποίησης δύο επιπέδων, παραπλήσια με τη μέθοδο που χρησιμοποιείται στο VP8, όπου τα μοντέλα πιθανότητας ανανεώνονται με βάση προηγούμενα στατιστικά κωδικοποίησης.



## 5 Android

Το Android είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance. Η Google εξαγόρασε το 2005 την εταιρία Android Inc που είχε ξεκινήσει την ανάπτυξη του το 2003. Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής hardware, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού. Οι προγραμματιστές αναπτύσσουν εφαρμογές για το Android σε μια τροποποιημένη έκδοση της Java. Το Android έγινε η κορυφαία πλατφόρμα διεθνώς για έξυπνα κινητά προς τα τέλη του 2010. Στα μέσα του 2012 υπήρχαν 400 εκατομμύρια συσκευές ενεργοποιημένες και 1 εκατομμύριο ενεργοποιήσεις καθημερινώς.

Η κύρια πλατφόρμα υλικού (hardware) ανάπτυξης είναι βασισμένη στην αρχιτεκτονική ARM. Παρόλο που έχει βασιστεί στον πυρήνα του Linux έχει ακολουθήσει ξεχωριστές αλλαγές αρχιτεκτονικής εκτός του τυπικού κύκλου ανάπτυξης του Linux.

Οι εκδόσεις του που έχουν κυκλοφορήσει μέχρι στιγμής παρουσιάζονται στον ακόλουθο πίνακα :

Έκδοση	Ημερομηνία	API level
1.5 Cupcake	30 Απριλίου 2009	3
1.6 Donut	15 Σεπτεμβρίου 2009	4
2.0, 2.1 Eclair	26 Οκτωβρίου 2009	7
2.2 Froyo	20 Μαΐου 2010	8
2.3.x Gingerbread	6 Δεκεμβρίου 2010	9-10
3.x.x Honeycomb	22 Φεβρουαρίου 2011	11-13
4.0.x Ice Cream Sandwich	19 Οκτωβρίου 2011	14-15
4.1.x Jelly Bean	10 Ιουλίου 2012	16

Σχήμα 18: Εκδόσεις λειτουργικού συστήματος Android



## 6 Υλοποίηση και Αποτελέσματα δοκιμών

### 6.1 Υλοποίηση εκτελέσιμων VP8 σε Windows και Android

Για την εκτέλεση δοκιμών χρησιμοποιήθηκε ο κώδικας αναφοράς libvpx (έκδοση 1.1.0)[15]. Για την πλατφόρμα των Windows δημιουργήθηκαν δυο εκτελέσιμα x86, ένα 32 bit και ένα 64 bit. Για την πλατφόρμα του Android χρησιμοποιήθηκε ένα εκτελέσιμο arm. Και στις δυο περιπτώσεις υποστηρίζεται η επιτάχυνση πολυμέσων SSE και SIMD NEON αντίστοιχα, όπως και η υποστήριξη multithreading.

### 6.2 Υλοποίηση H.264

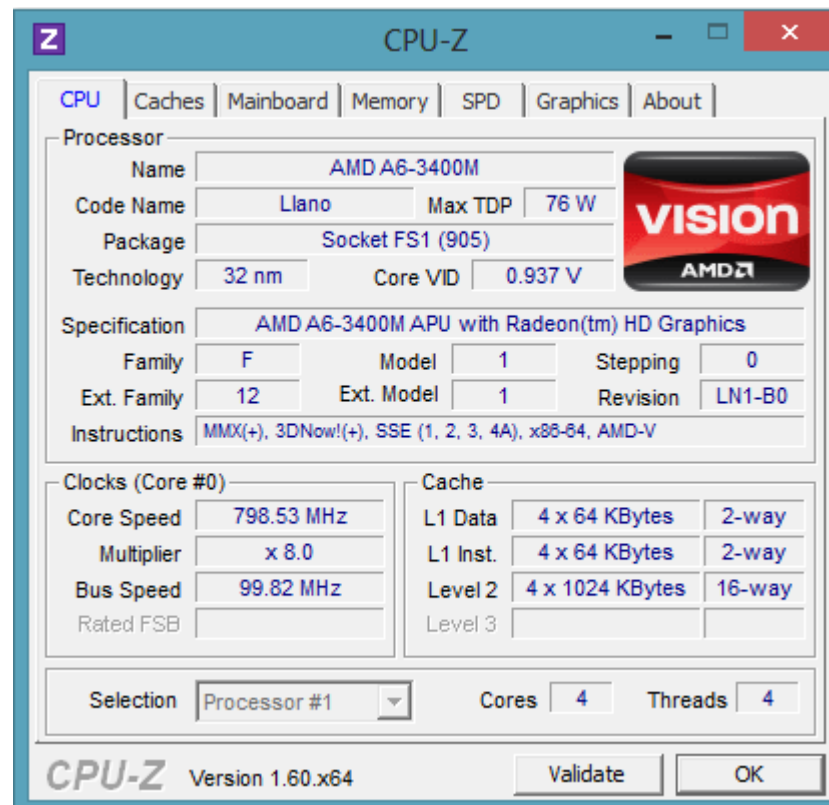
Για την εκτέλεση δοκιμών χρησιμοποιήθηκε ο codec αναφοράς JM Reference Software [16] (έκδοση 18.4, Ημερομηνία έκδοσης 2012-08-04). Δημιουργήθηκαν δυο εκτελέσιμα x86 για την πλατφόρμα των Windows , ένα 32 bit και ένα 64 bit. Και στις δυο περιπτώσεις υποστηρίζεται η επιτάχυνση πολυμέσων SSE αλλά απουσιάζει η υποστήριξη multithreading.

### 6.3 Υλοποίηση εργαλείων

Για την εκτέλεση των μετρήσεων ποιότητας χρησιμοποιήθηκε ένα εργαλείο (YUVDIFF) σε κώδικα C για την μέτρηση της ποιότητας σε PSNR καθώς και κώδικας σε MATLAB για τη διαφορά εμβαδών PSNR/Bitrate.

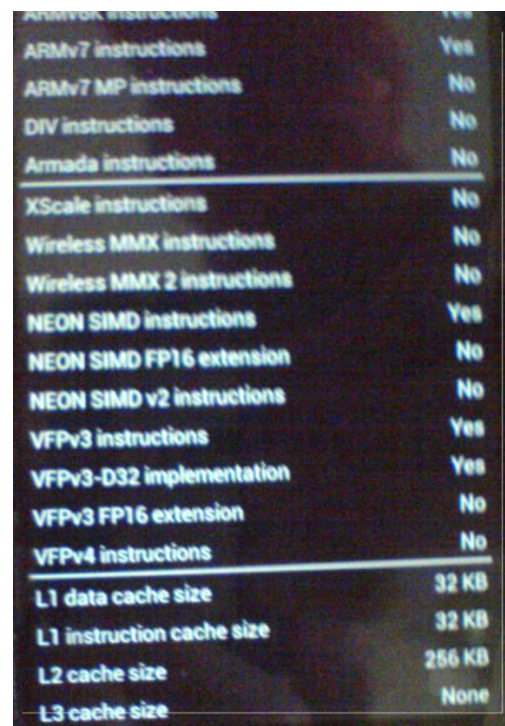
### 6.4 Προδιαγραφές συστημάτων εκτέλεσης

Για την εκτέλεση των δοκιμών χρησιμοποιήθηκε ένας προσωπικός υπολογιστής και ένα κινητό. Ο υπολογιστής αποτελείται από έναν επεξεργαστή AMD A6-3400m , με μνήμη 8GB DDR3 1333MHz με λειτουργικό σύστημα Window-7 64-bit.



Σχήμα 18: Χαρακτηριστικά επεξεργαστή προσωπικού υπολογιστή

Το κινητό είναι το Xperia Neo V με επεξεργαστή Snapdragon S2 (MSM8255) τεχνολογίας ARM-Cortex A8 (dual) και μνήμη LPDDR2 μεγέθους 512MB συχνότητας 500 MHz.



Σχήμα 19: Χαρακτηριστικά επεξεργαστή κινητού

Πιο αναλυτικά τα χαρακτηριστικά των επεξεργαστών είναι:

	AMD Fusion APU x86	Snapdragon S2 (MSM8255)
Αρχιτεκτονική	x86-64	ARMv7
Συχνότητα CPU	1.4 GHz	1.0 GHz
Αριθμός Πυρήνων	4	1
Decode		2-wide
Βάθος Pipeline		10 stages
Multimedia Extensions	MMX,SSE, SSE2, SSE3, SSE4a	NEON
FPU		VFPv3 (pipelined)
L1 data cache	2-way set associative 64KB (ανά πυρήνα)	2-way set associative 32KB
L1 instruction cache	2-way set associative 64KB (ανά πυρήνα)	2-way set associative 32KB
L2 cache	16 way 1024KB (ανά πυρήνα)	256KB

Σχήμα 20: Σύγκριση χαρακτηριστικών επεξεργαστών

## 6.5 Αποτελέσματα Μετρήσεων

### 6.5.1 Dataset properties

Για την εκτέλεση των μετρήσεων χρησιμοποιήθηκαν τα ακόλουθα video από το Video Quality Experts Group (VQEG) [9], με τα εξής χαρακτηριστικά

Όνομα αρχείου	Ανάλυση	Τύπος πηγής	subsampling scheme	Σύνολο καρέ
src13_ref_720x480_420	720x480	Interlaced	4:2:0	260
src14_ref_720x480_420	720x480	Interlaced	4:2:0	260
src15_ref_720x480_420	720x480	Interlaced	4:2:0	260
src16_ref_720x480_420	720x480	Progressive	4:2:0	260
src17_ref_720x480_420	720x480	Progressive	4:2:0	260
src18_ref_720x480_420	720x480	Interlaced	4:2:0	260
src19_ref_720x480_420	720x480	Interlaced	4:2:0	260
Src20_ref_720x480_420	720x480	Interlaced	4:2:0	260
Src21_ref_720x480_420	720x480	Interlaced	4:2:0	260
Src22_ref_720x480_420	720x480	Interlaced	4:2:0	260

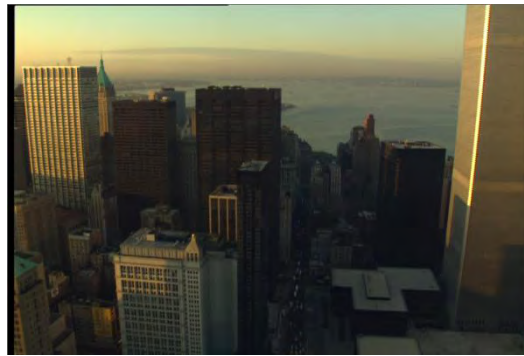
Σχήμα 21: Χαρακτηριστικά των video του Dataset



src13\_ref\_\_720x480\_420



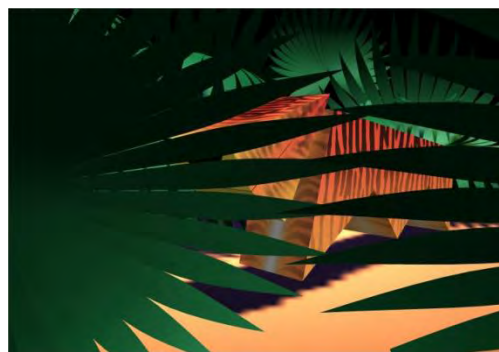
src14\_ref\_\_720x480\_420



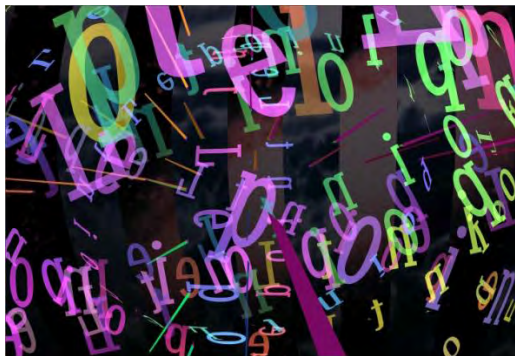
src15\_ref\_\_720x480\_420



src16\_ref\_\_720x480\_420



src17\_ref\_\_720x480\_420



src18\_ref\_\_720x480\_420



src19\_ref\_\_720x480\_420



src20\_ref\_\_720x480\_420





Src21\_ref\_720x480\_420



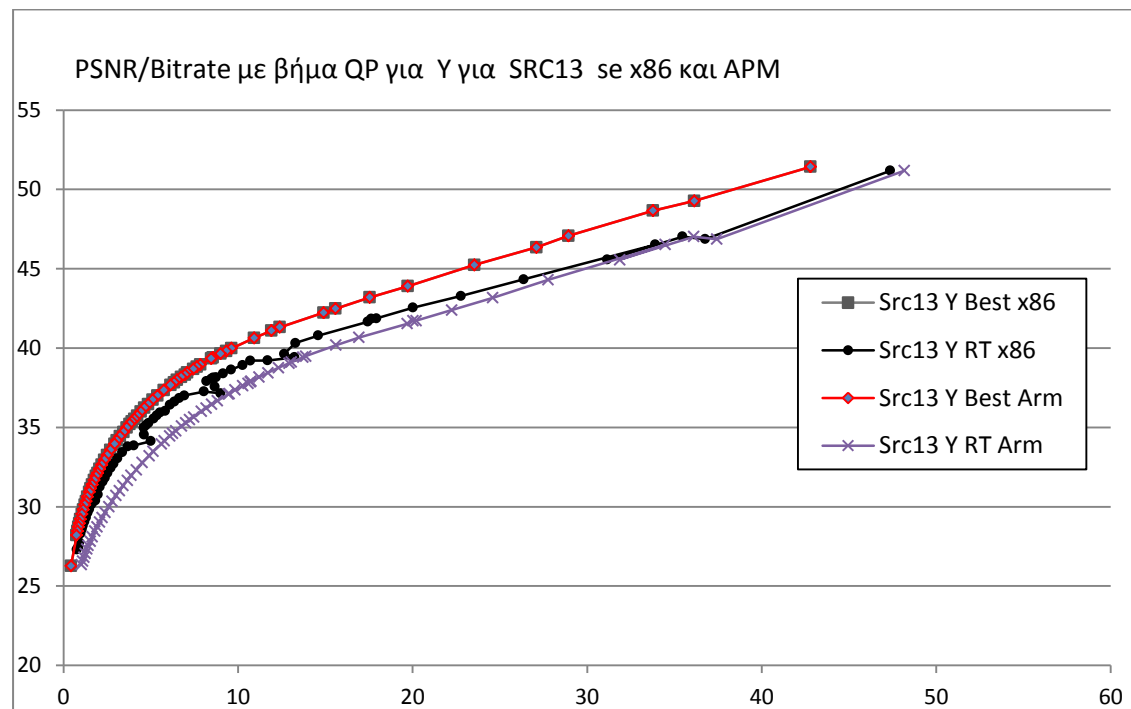
src22\_ref\_720x480\_420



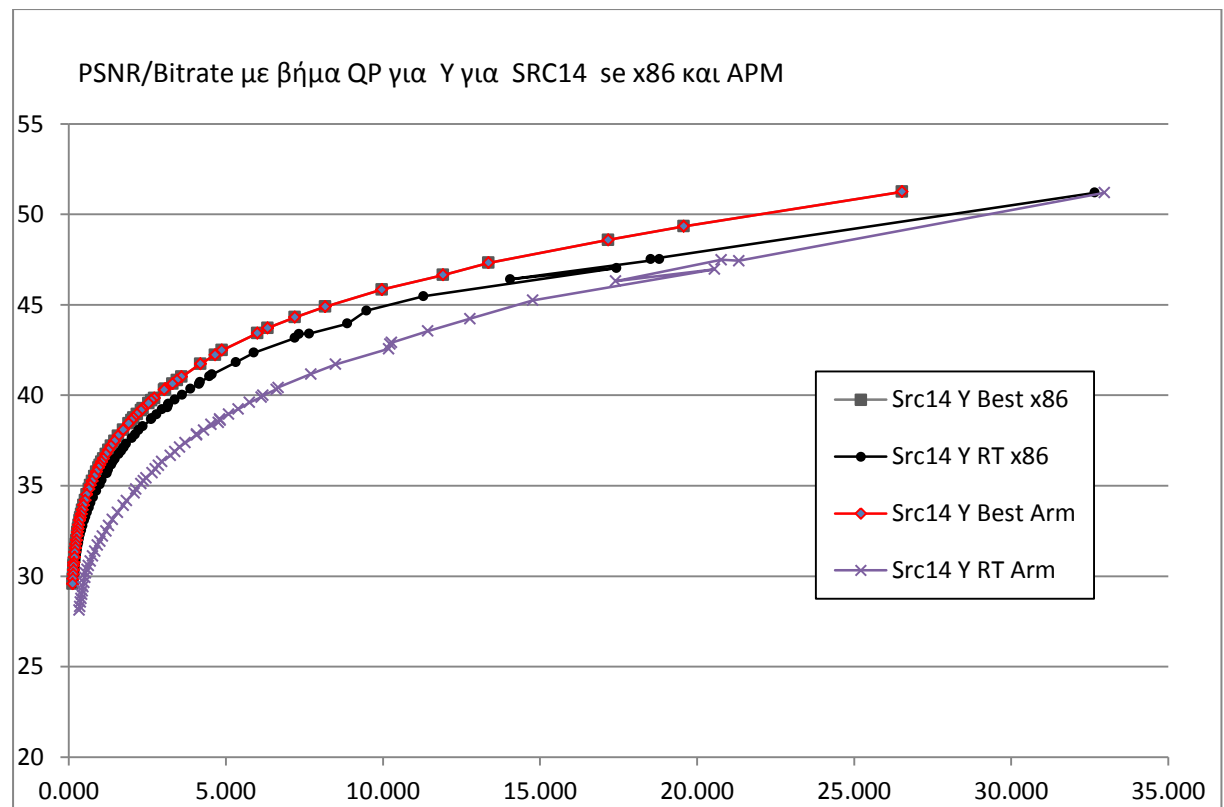
## 6.5.2 Αποδοτικότητα encoders

### 6.5.2.1 VP8 Λόγος PSNR/Bitrate

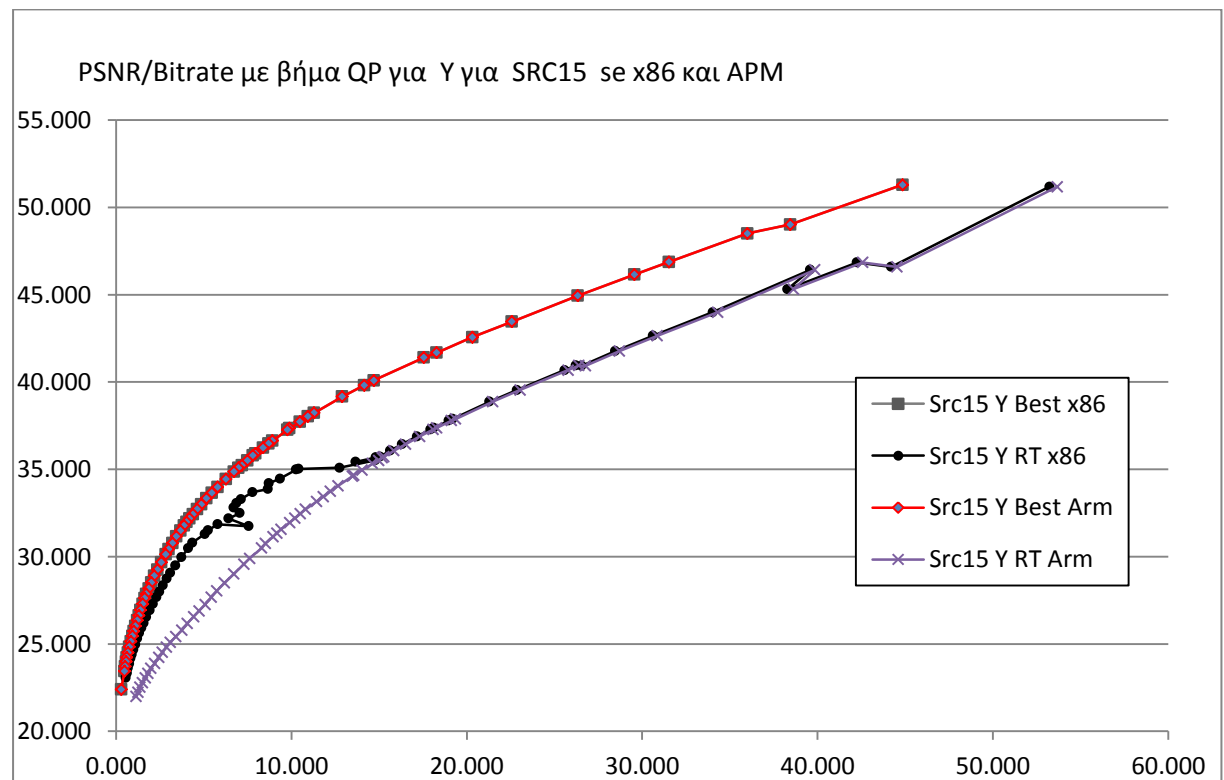
Παρακάτω φαίνονται τα διαγράμματα PSNR/Bitrate για όλα τα sources με συμπίεση VP8 σε αρχιτεκτονική x86 και ARM. Παρατηρούμε ότι για ποιότητα Best οι γραφικές παραστάσεις ταυτίζονται.



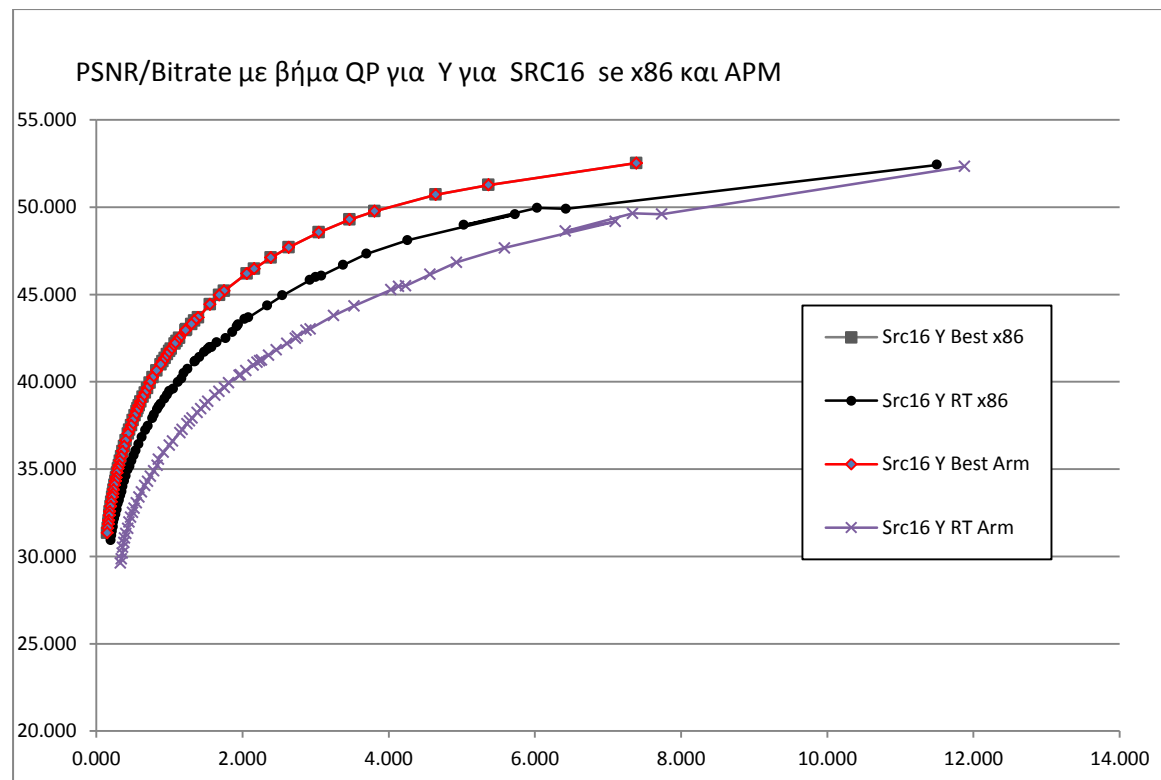
Σχήμα 22.1.13: Λόγος PSNR/Bitrate με βήμα QP για τη συνιστώσα Y για SRC13 σε x86 και APM



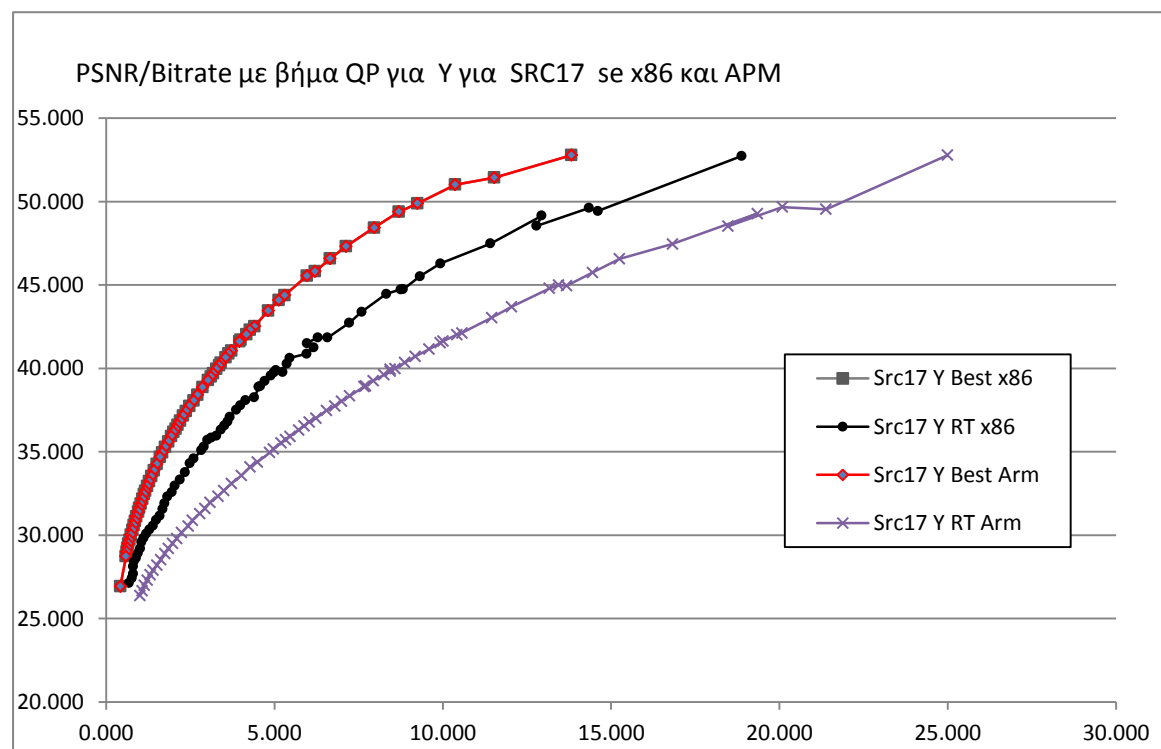
Σχήμα 22.1.14: Λόγος PSNR/Birate με βήμα QP για τη συνιστώσα Y για SRC14 σε x86 και APM



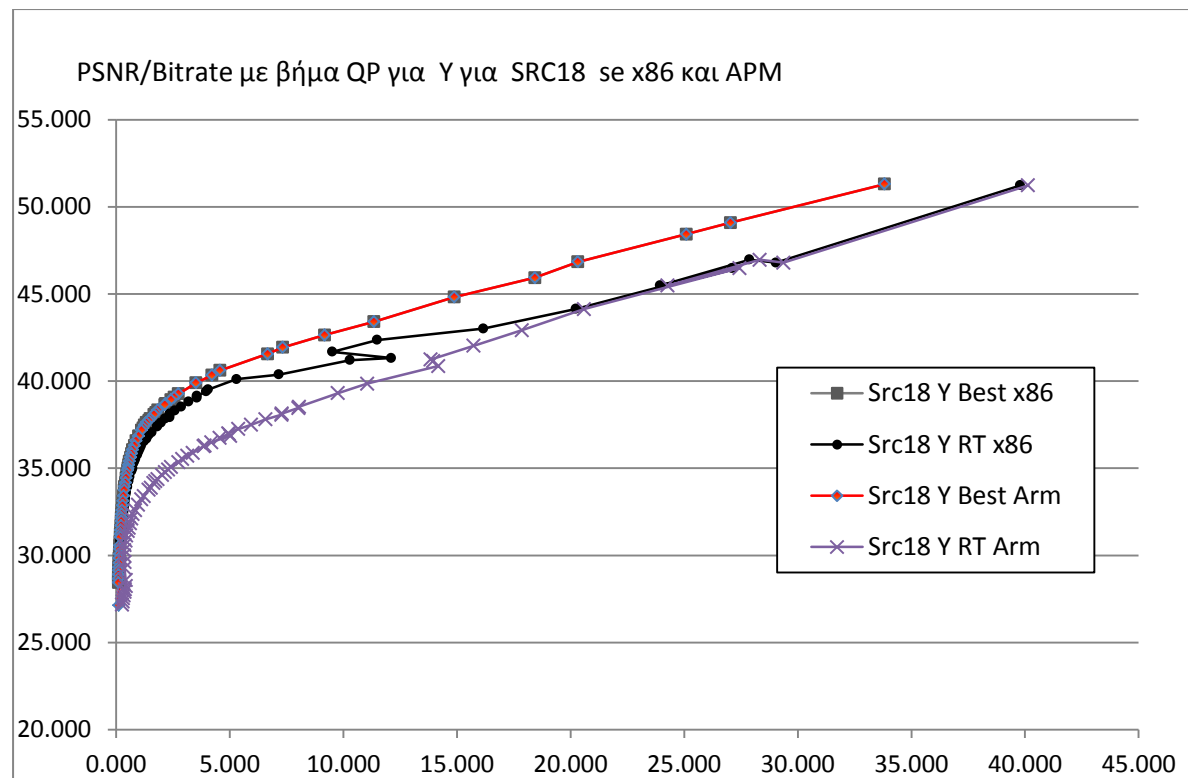
Σχήμα 22.1.15: Λόγος PSNR/Birate με βήμα QP για τη συνιστώσα Y για SRC15 σε x86 και APM



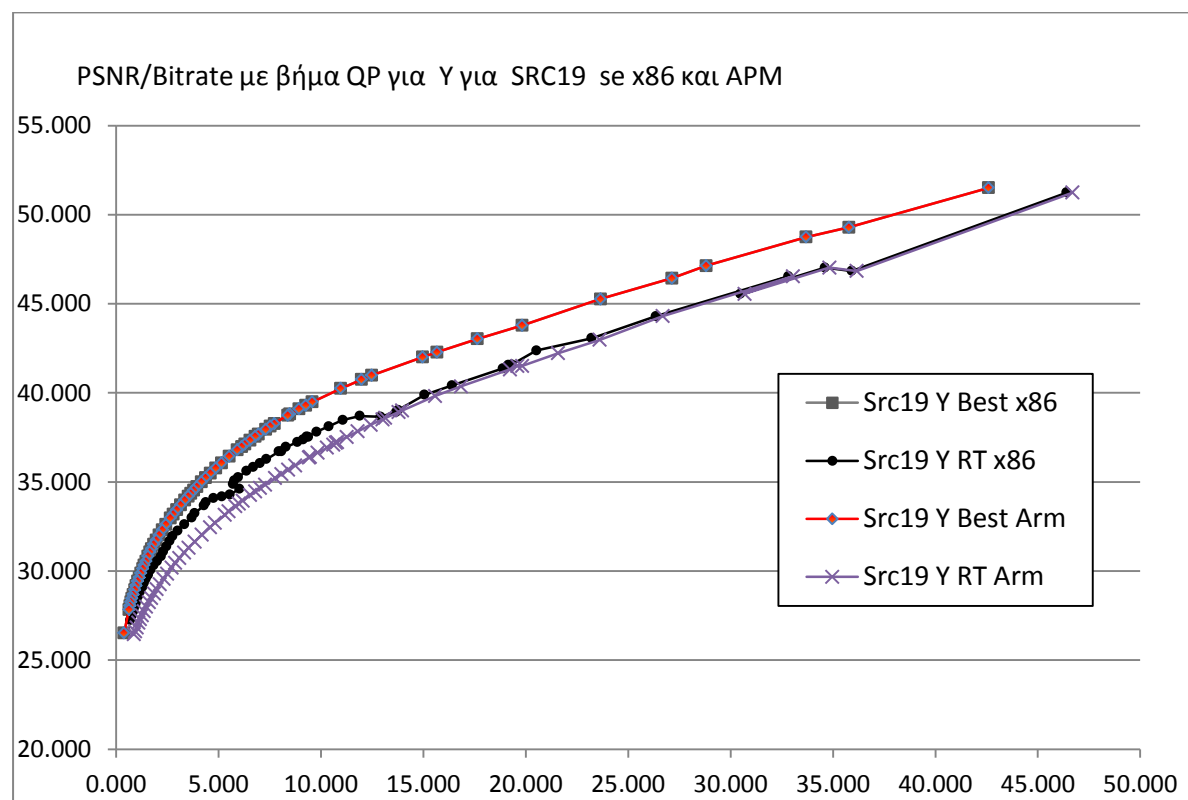
Σχήμα 22.1.16: Λόγος PSNR/Bitrate με βήμα QP για τη συνιστώσα Y για SRC16 σε x86 και APM



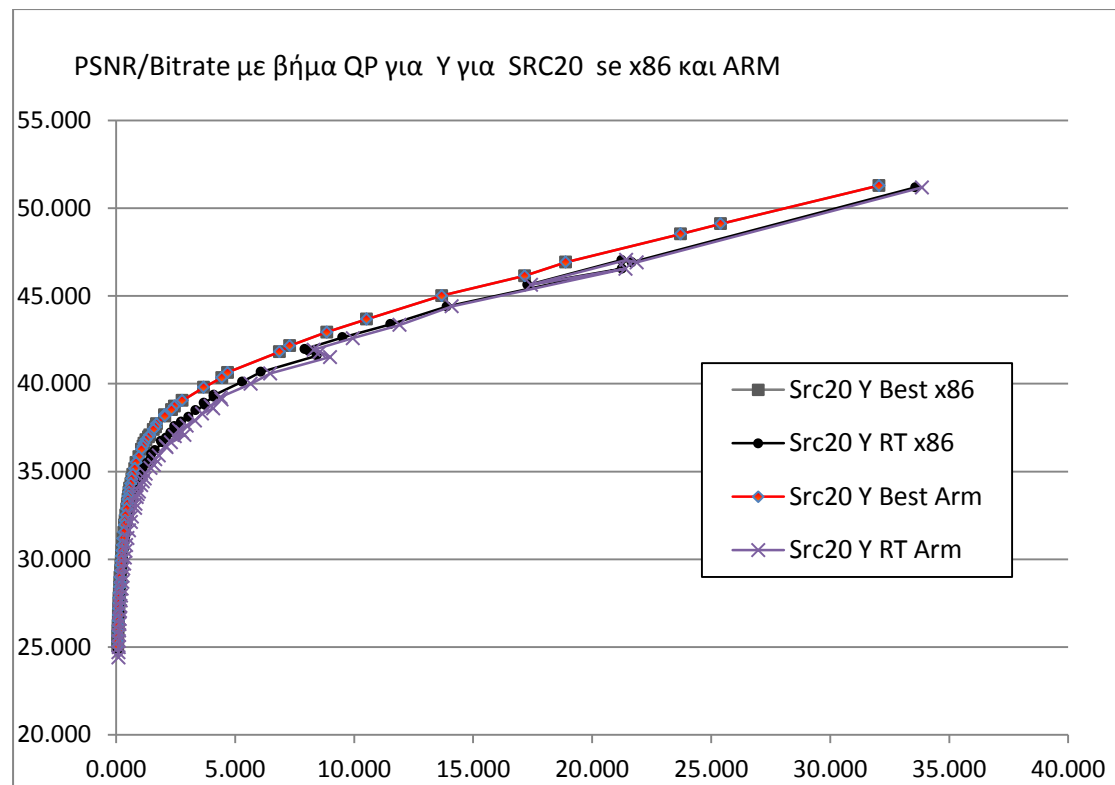
Σχήμα 22.1.17: Λόγος PSNR/Bitrate με βήμα QP για τη συνιστώσα Y για SRC17 σε x86 και APM



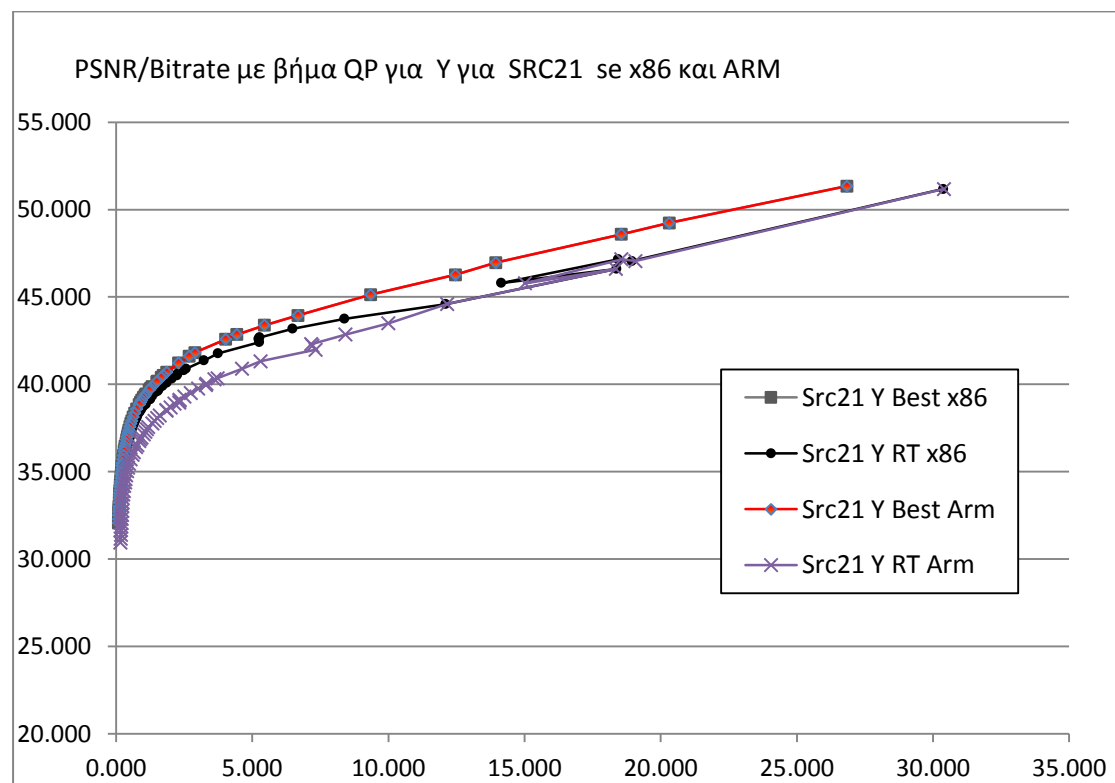
Σχήμα 22.1.18: Λόγος PSNR/Bitrate με βήμα QP για τη συνιστώσα Y για SRC18 σε x86 και APM



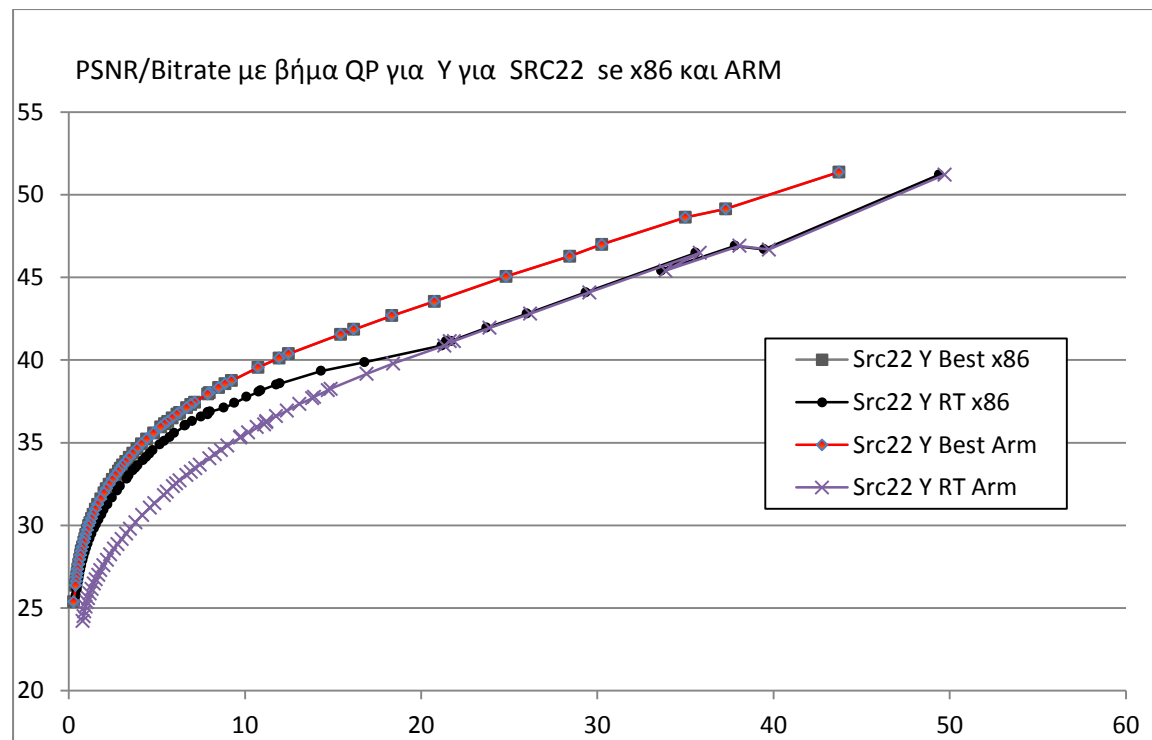
Σχήμα 22.1.19: Λόγος PSNR/Bitrate με βήμα QP για τη συνιστώσα Y για SRC19 σε x86 και APM



Σχήμα 22.1.20: Λόγος PSNR/Bitrate με βήμα QP για τη συνιστώσα Y για SRC20 σε x86 και APM



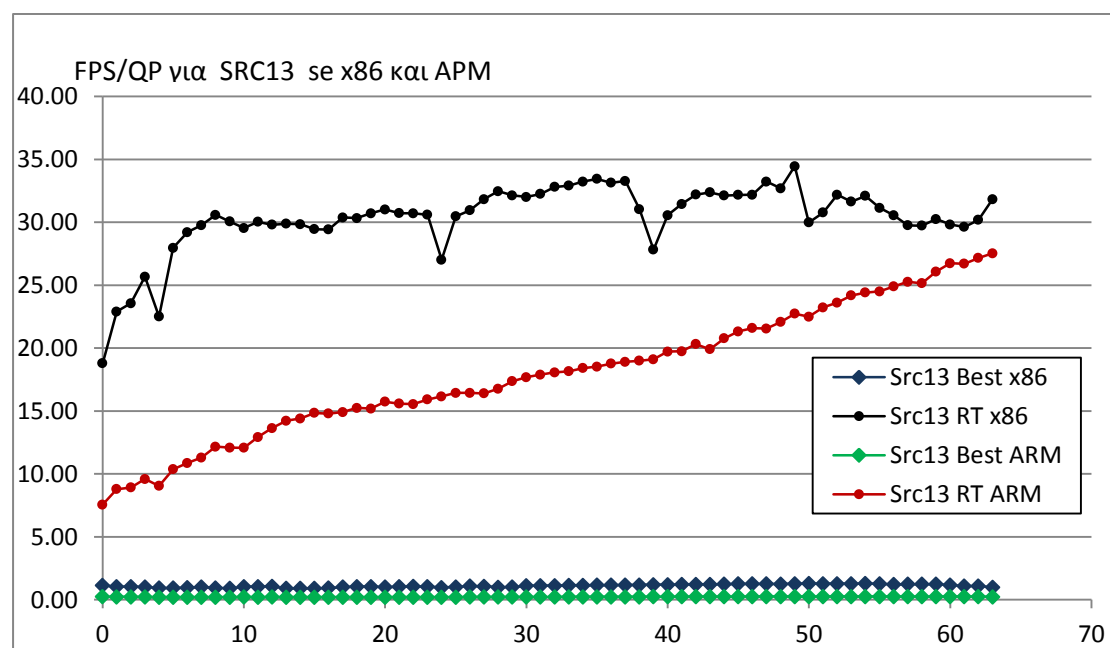
Σχήμα 22.1.20: Λόγος PSNR/Bitrate με βήμα QP για τη συνιστώσα Y για SRC21 σε x86 και APM



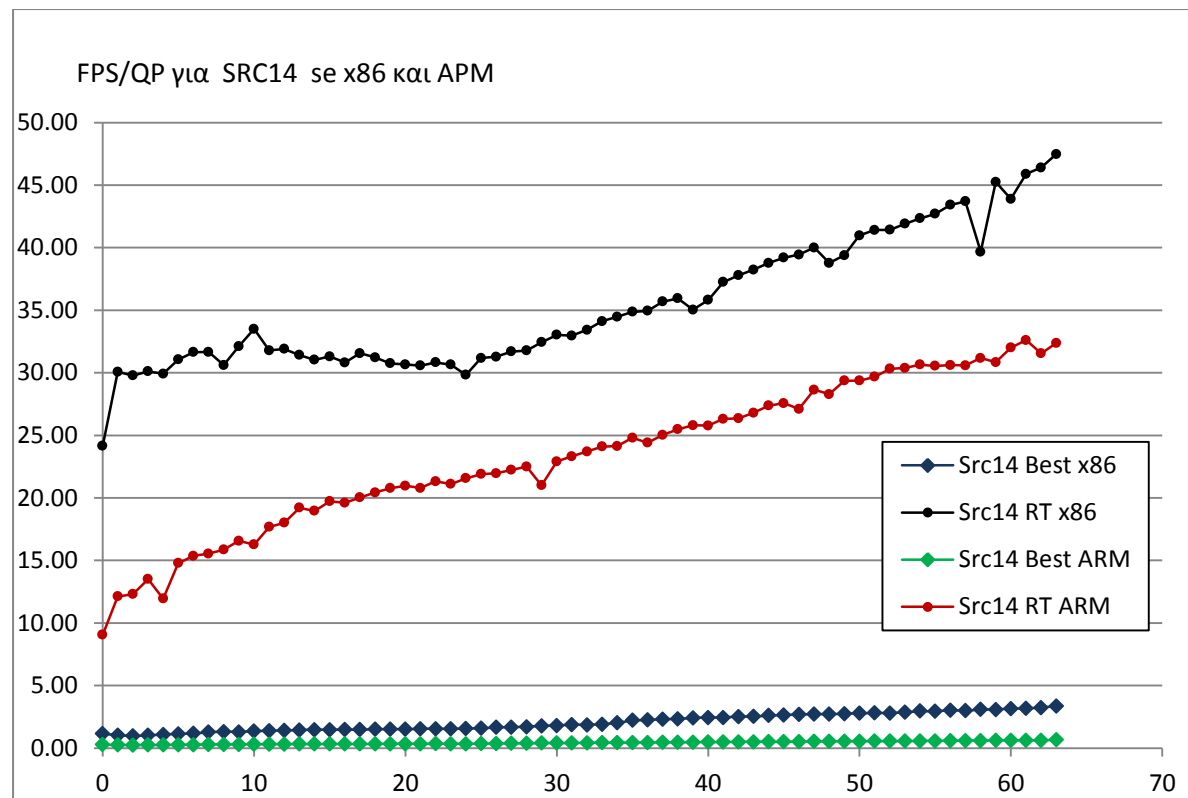
Σχήμα 22.1.20: Λόγος PSNR/Bitrate με βήμα QP για τη συνιστώσα Y για SRC22 σε x86 και ARM

#### 6.5.2.2 Ταχύτητα κωδικοποίησης VP8 σε x86 και ARM

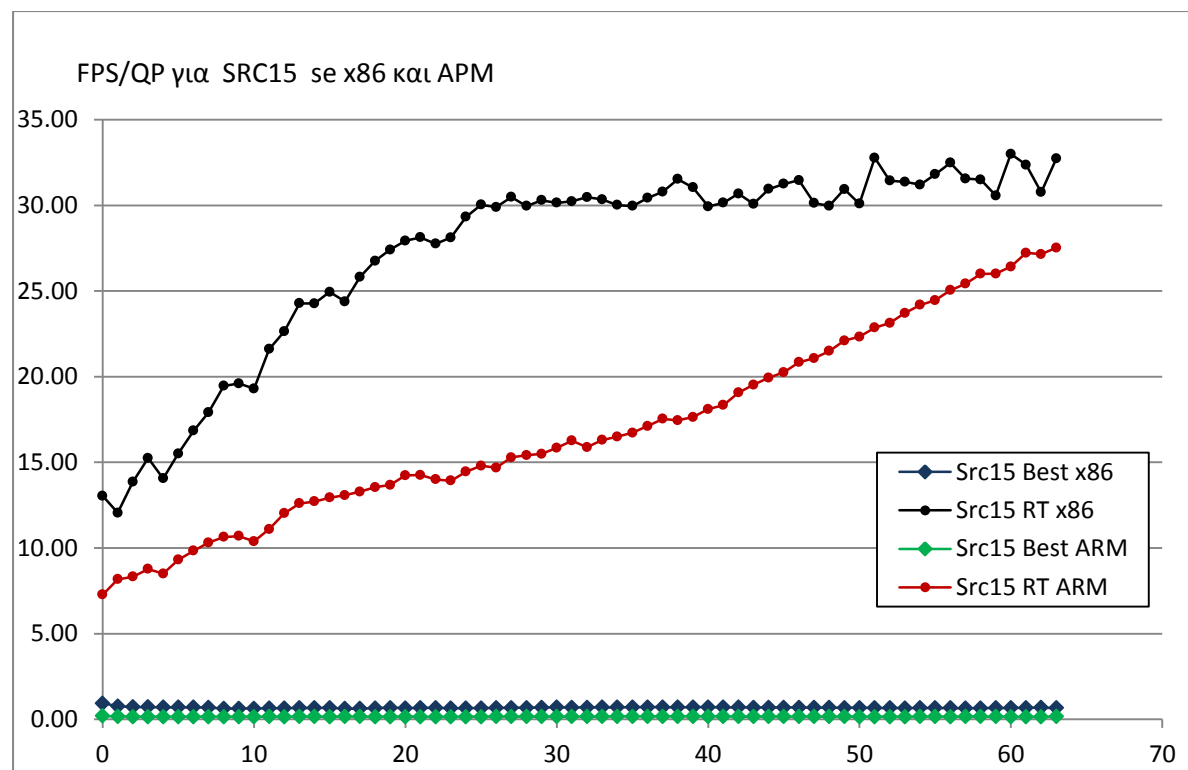
Παρακάτω φαίνεται η ταχύτητα κωδικοποίησης σε πλατφόρμα x86 και ARM και για ποιότητα best και realtime για κάθε βίντεο ξεχωριστά.



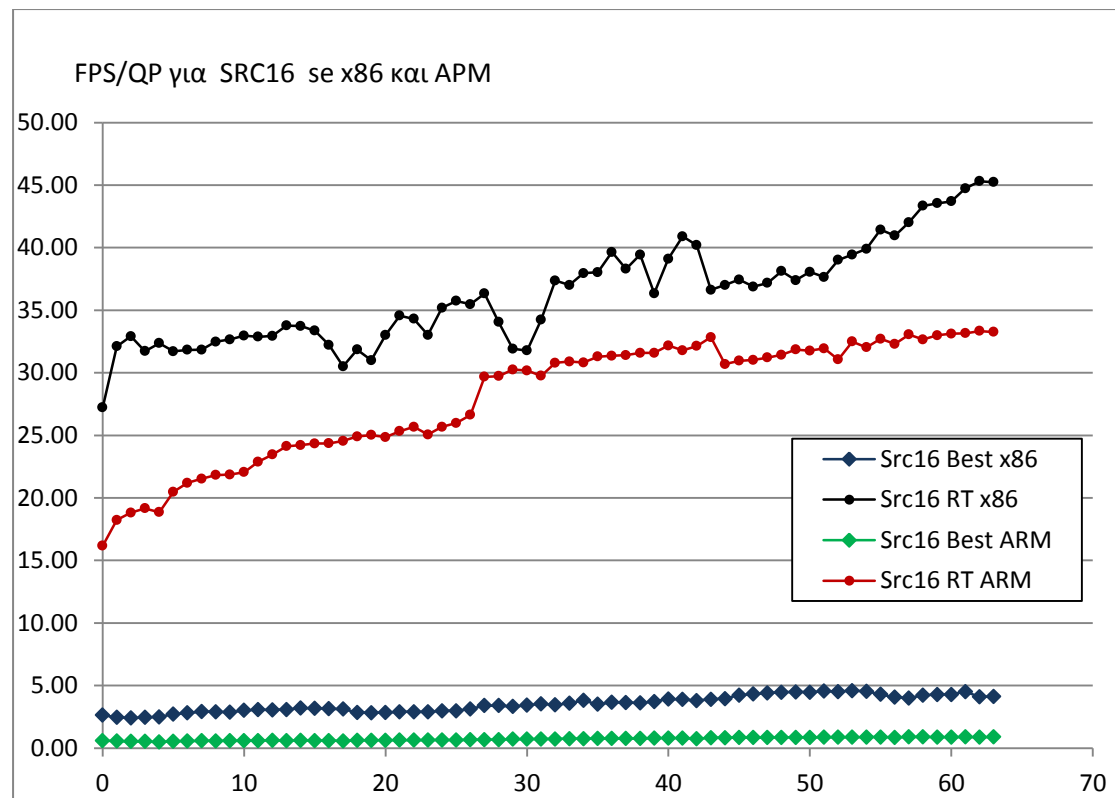
Σχήμα 22.4.13: FPS/QP για src13 με VP8



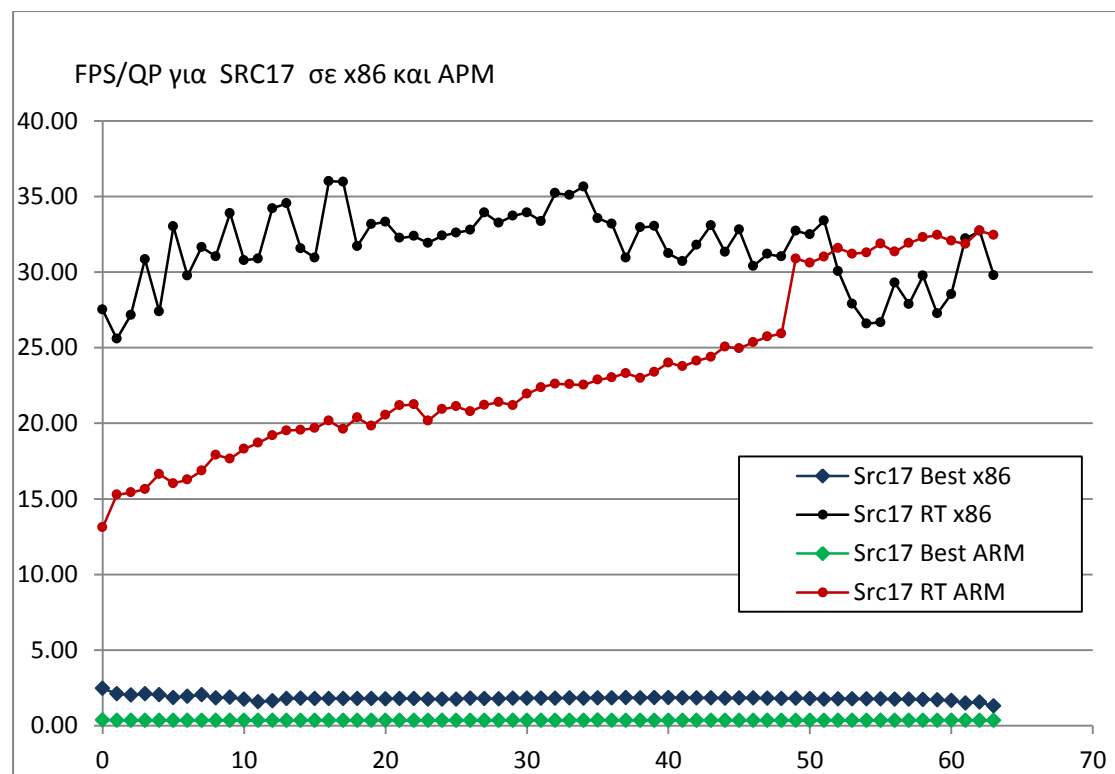
Σχήμα 22.4.14: FPS/QP για src14 με VP8



Σχήμα 22.4.15: FPS/QP για src15 με VP8

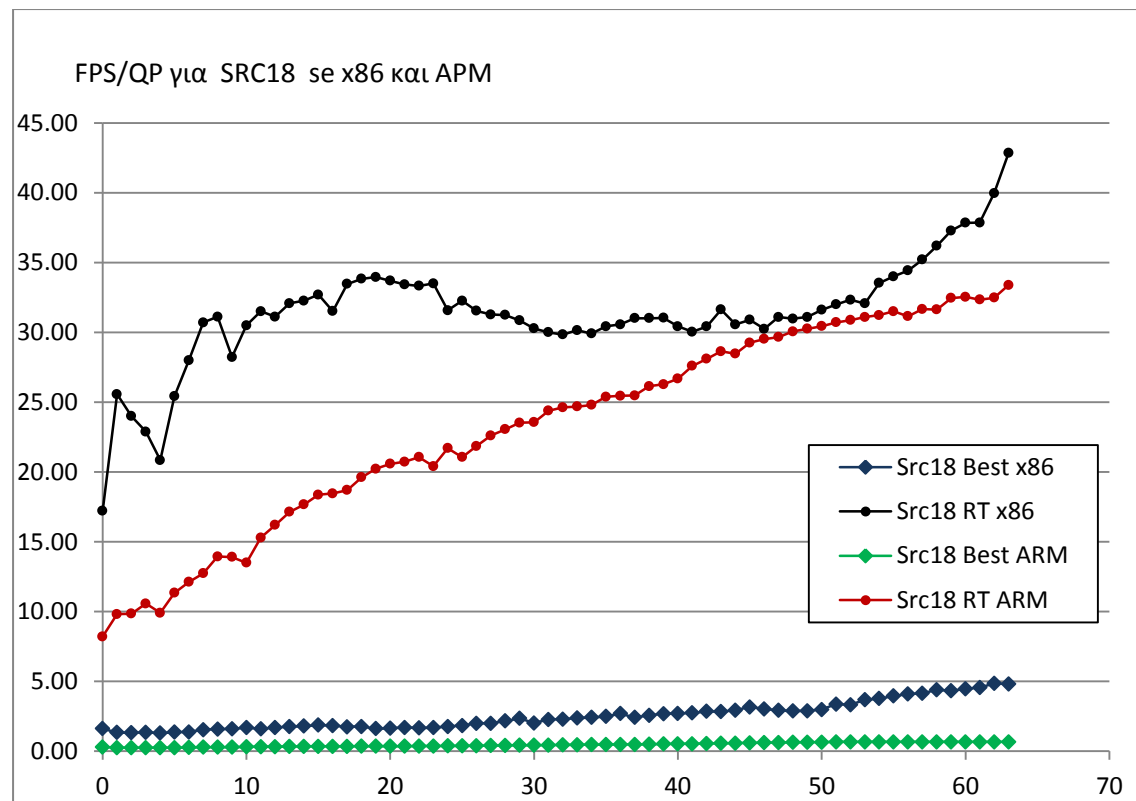


Σχήμα 22.4.16: FPS/QP για src16 με VP8

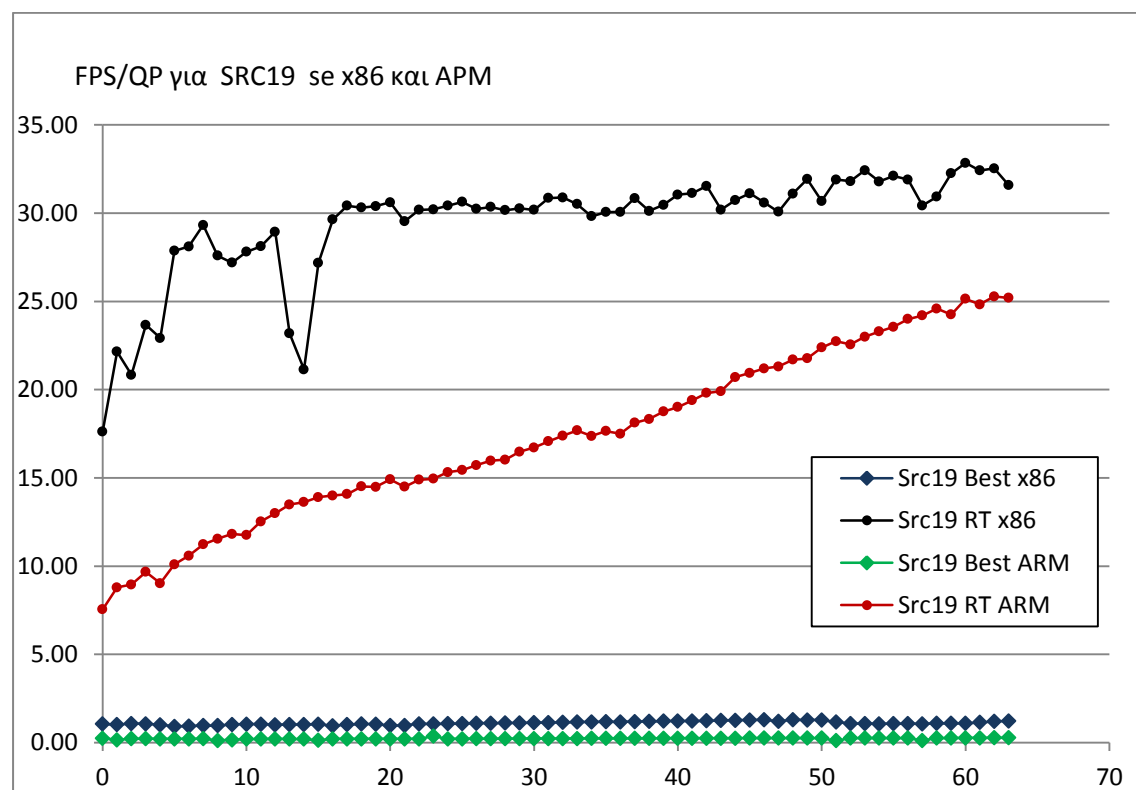


Σχήμα 22.4.17: FPS/QP για src17 με VP8

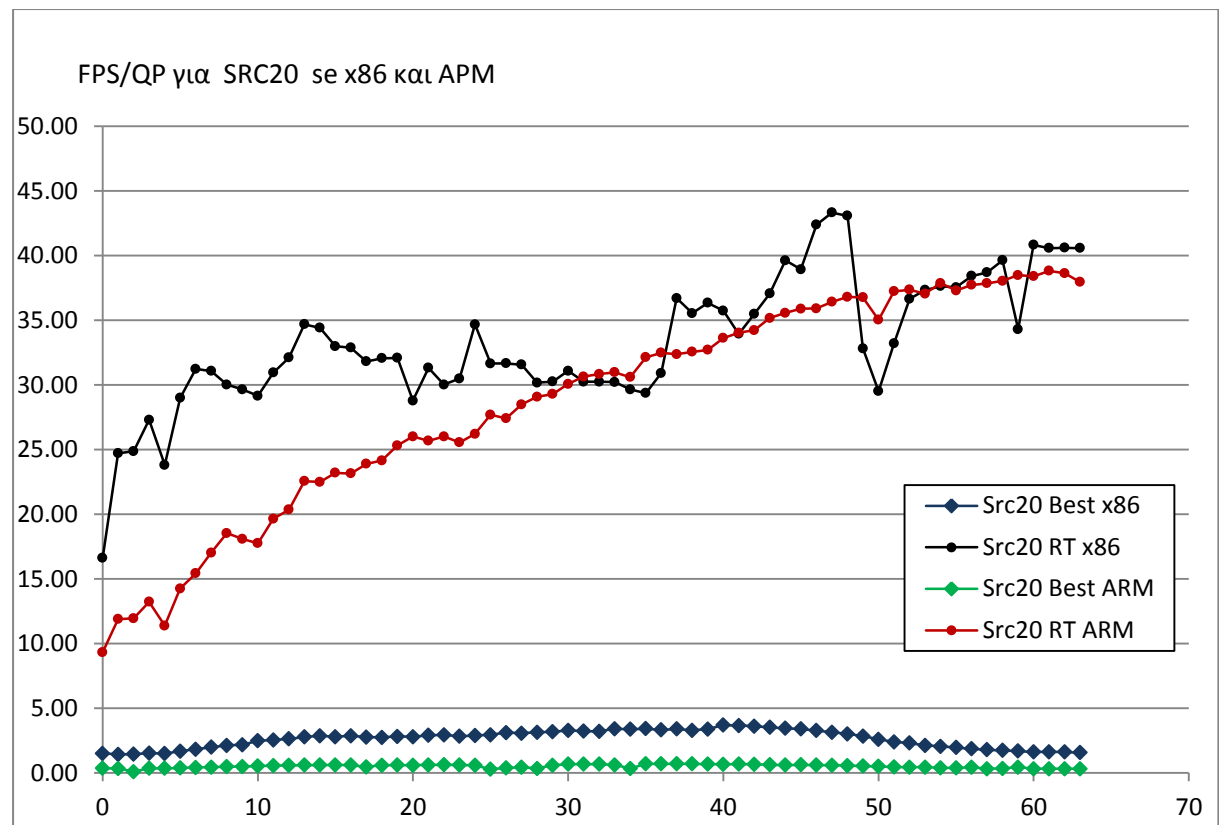




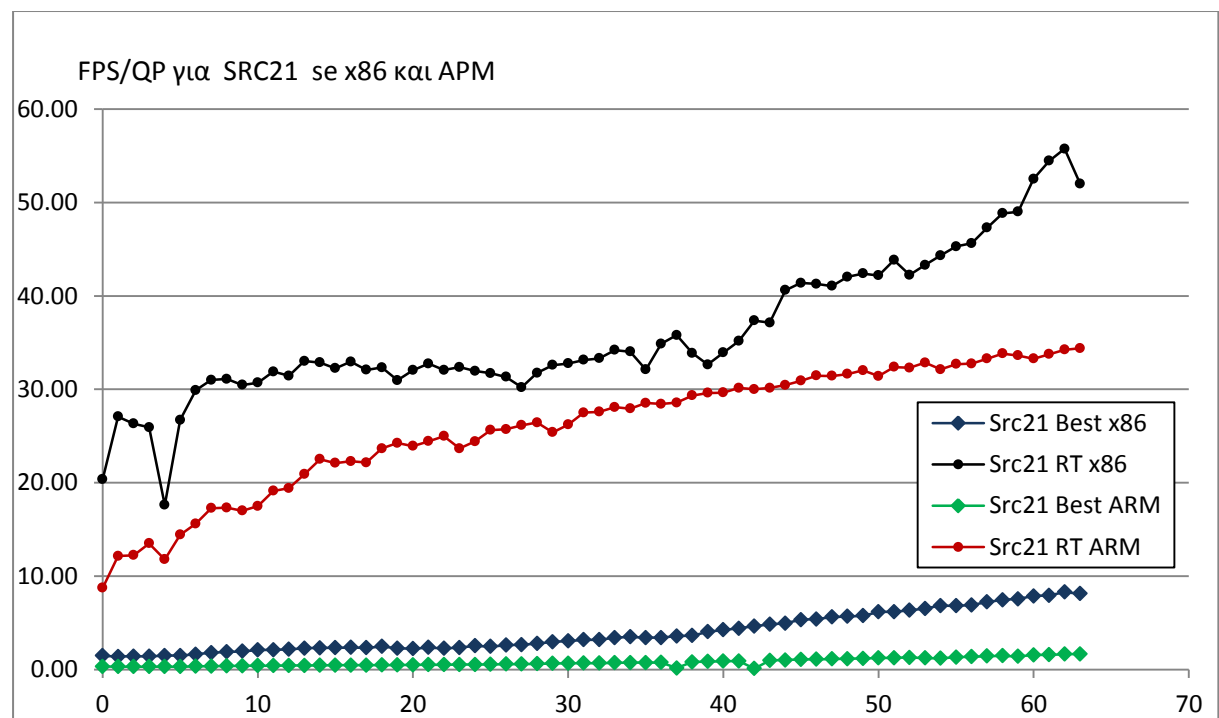
Σχήμα 22.4.18: FPS/QP για src18 με VP8



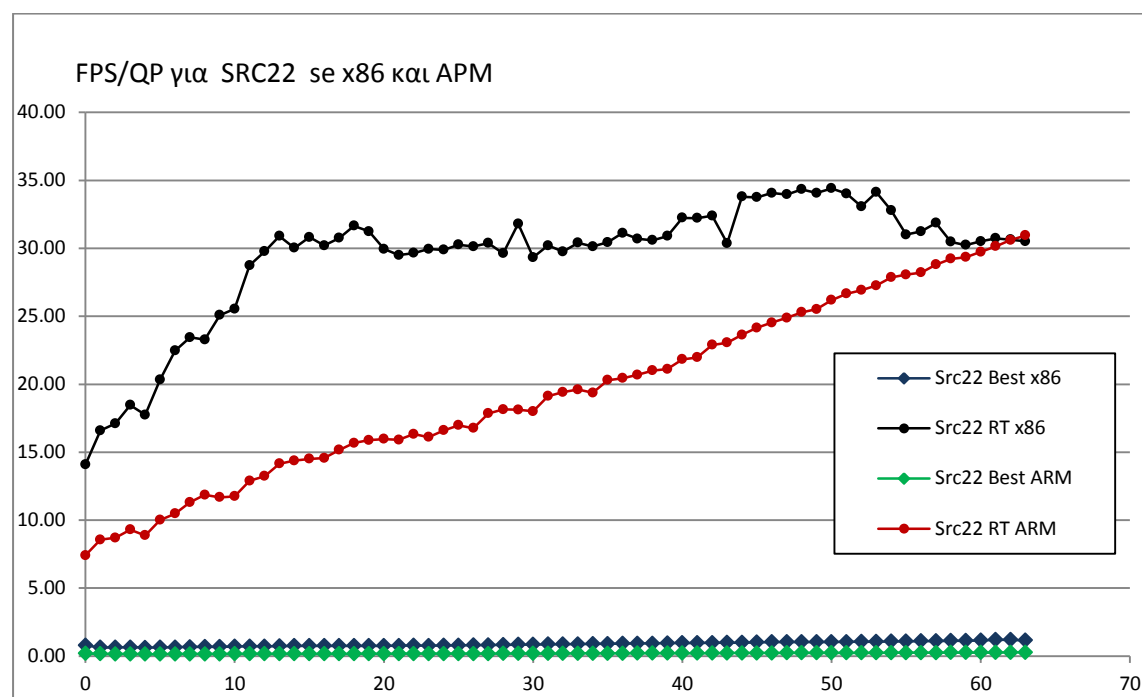
Σχήμα 22.4.19: FPS/QP για src19 με VP8



Σχήμα 22.4.20: FPS/QP για src20 με VP8

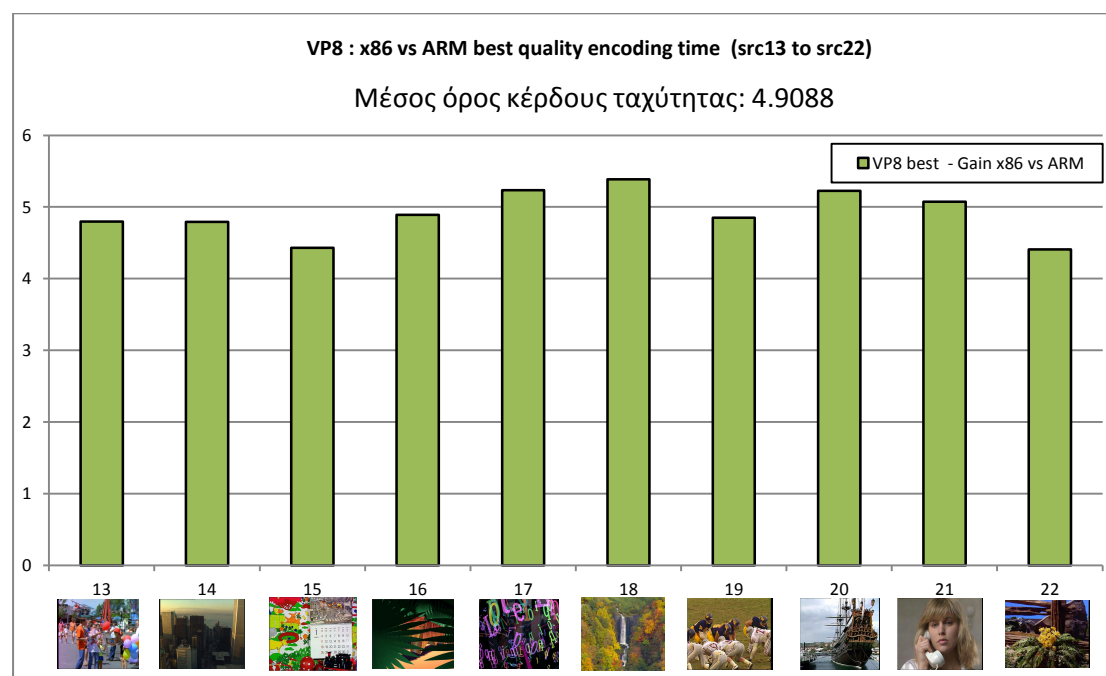


Σχήμα 22.4.21: FPS/QP για src21 με VP8



Σχήμα 22.4.22: FPS/QP για src22 με VP8

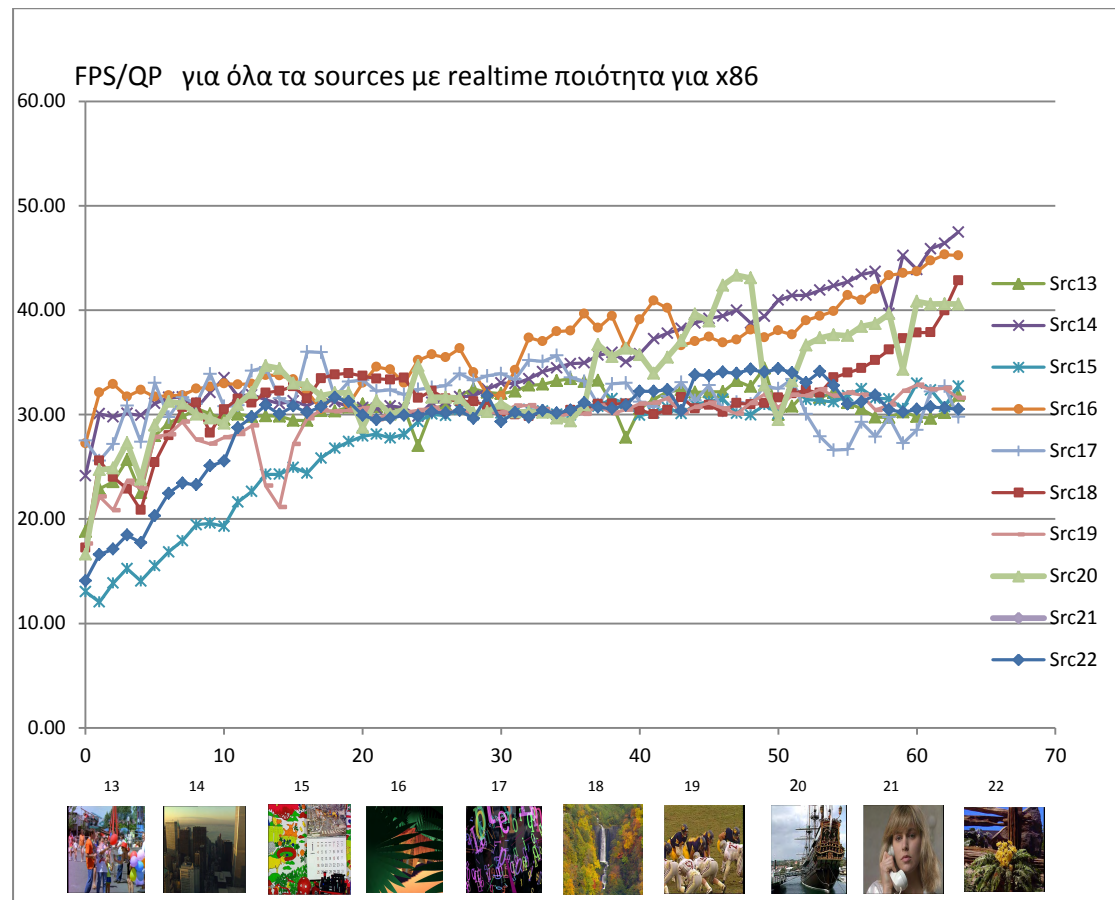
Στην συνέχεια παρουσιάζεται το συνολικό διάγραμμα διαφοράς χρόνου κωδικοποίησης μεταξύ αρχιτεκτονικής x86 και ARM, όταν έχουμε ποιότητα best.



Σχήμα 22.5: Χρόνος κωδικοποίησης VP8 για πλατφόρμες x86 και ARM για ποιότητα best.

Όπως είδαμε και στα διαγράμματα ποιότητας σε ποιότητα best έχουμε το ίδιο ποιοτικά αρχείο, ωστόσο αυτό για να επιτευχθεί αυτό έχουμε το ανάλογο κόστος σε ταχύτητα. Στα συγκεκριμένα συστήματα δοκιμών ο υπολογιστής εκτέλεσε την κωδικοποίηση κατά μέσο όρο

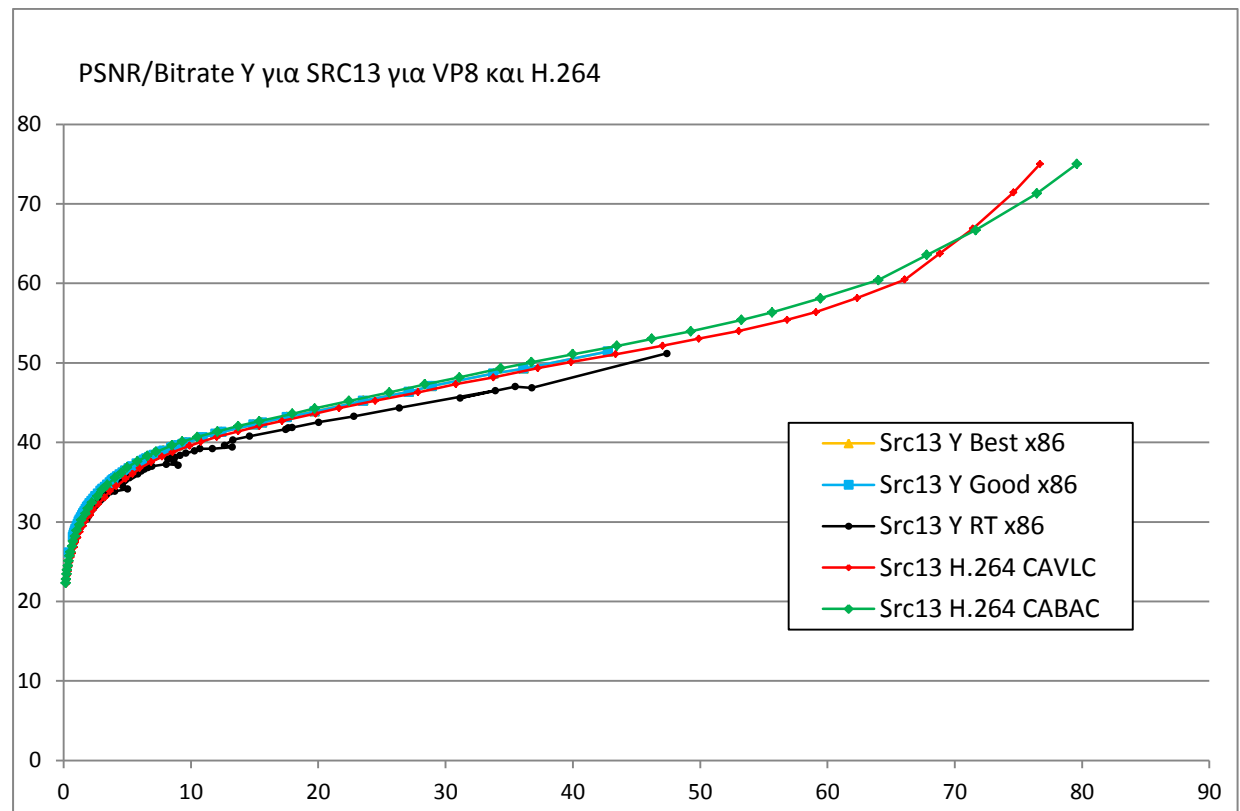
4.9088 πιο γρήγορα απ' ότι το κινητό. Αυτό οφείλεται κυρίως στην υψηλότερη συχνότητα λειτουργία του και την μεγαλύτερη μνήμη cache.



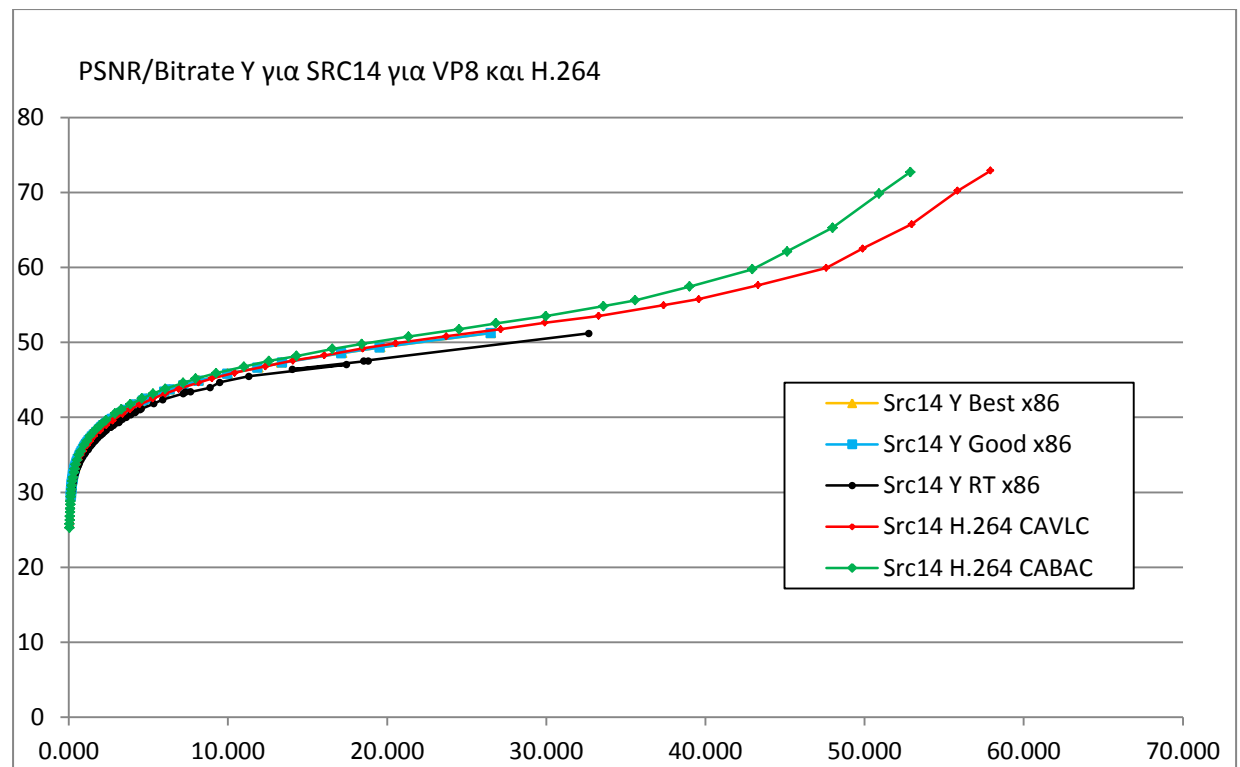
Σχήμα 22.6: FPS/QP για όλα τα βίντεο με realtime ποιότητα του VP8 για x86

### 6.5.2.3 VP8 vs H.264

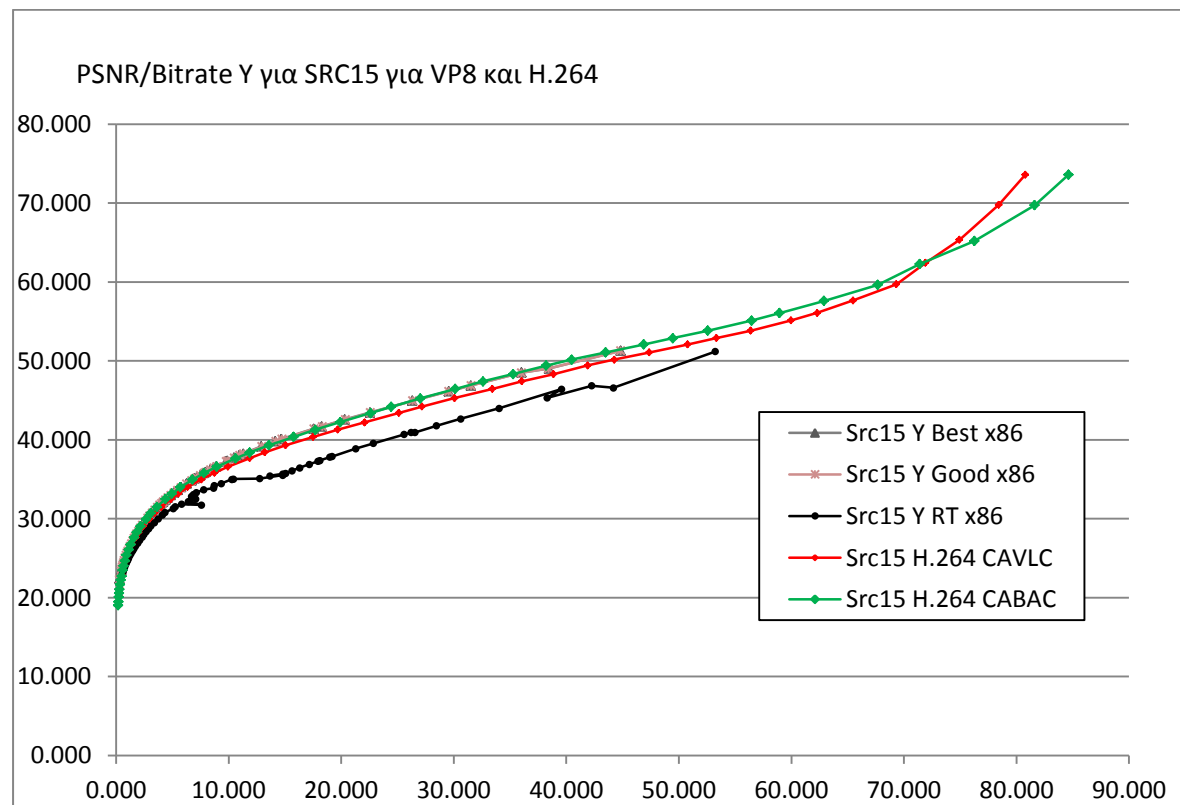
Ακολουθούν διαγράμματα σύγκρισης PSNR/Bitrate για τη συνιστώσα Y μεταξύ VP8 και H.264 ξεχωριστά για κάθε βίντεο



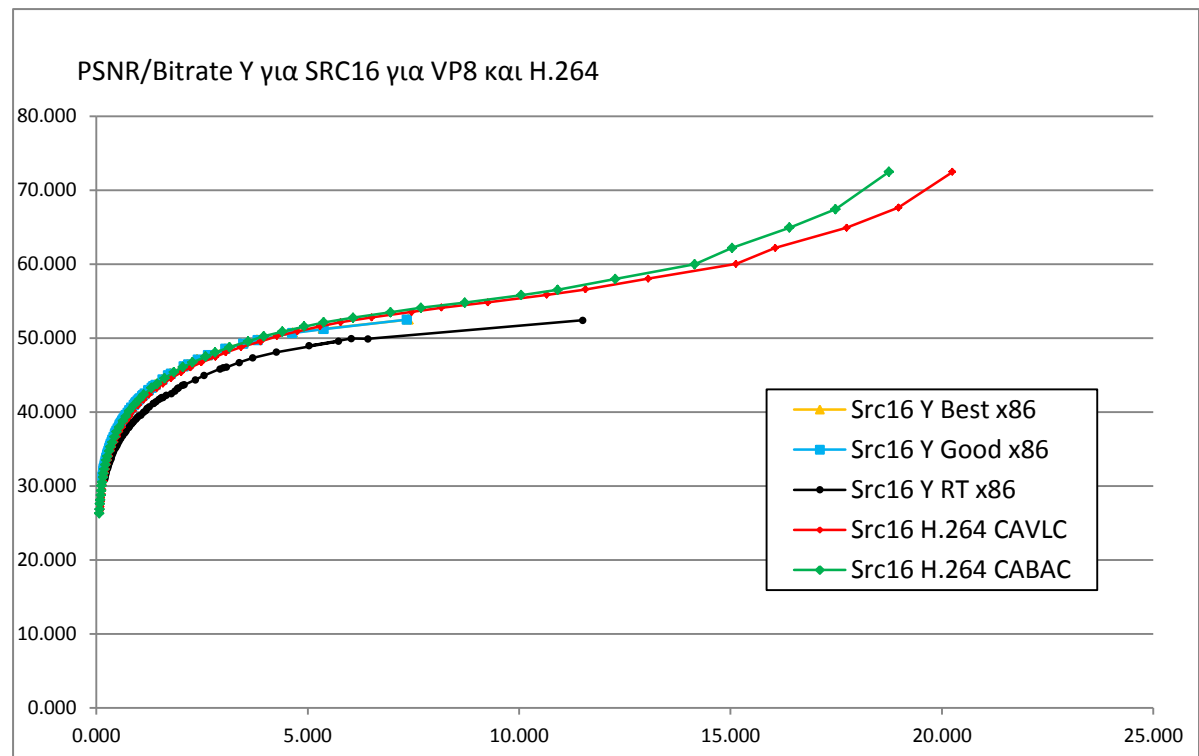
Σχήμα 22.7.13: PSNR/Bitrate Y για SRC13 για VP8 και H.264



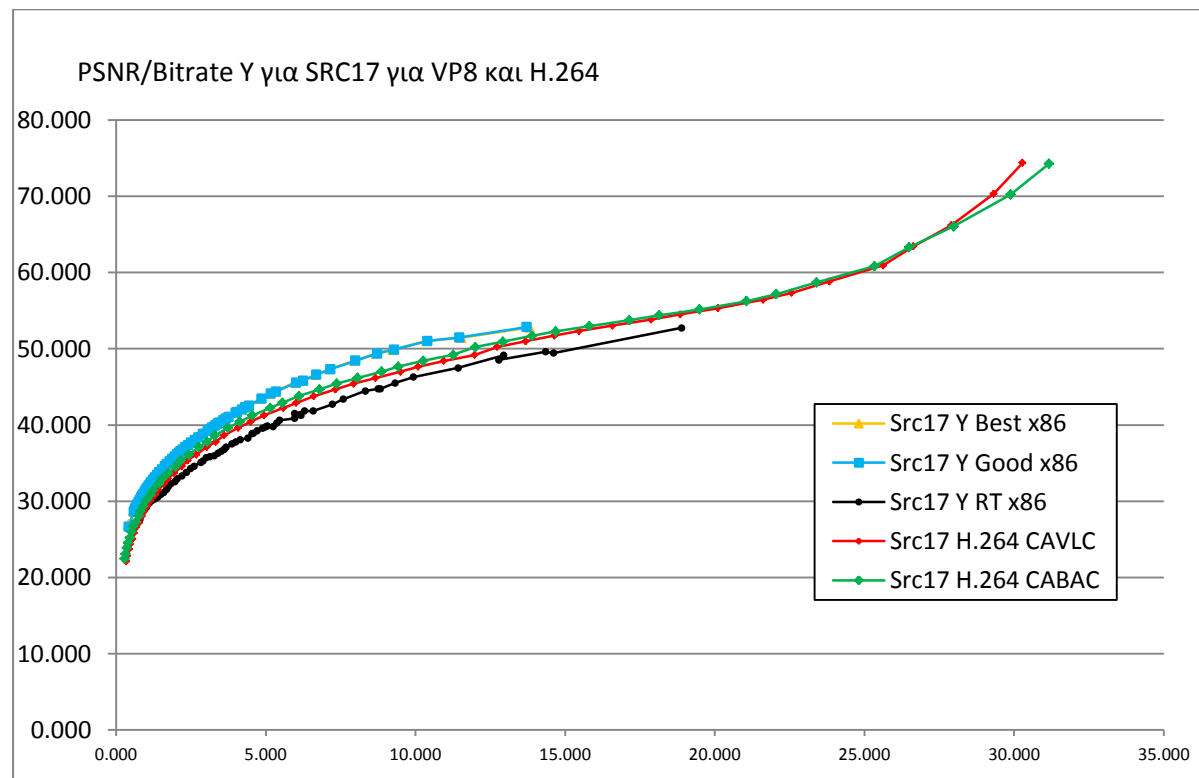
Σχήμα 22.7.14: PSNR/Bitrate Y για SRC14 για VP8 και H.264



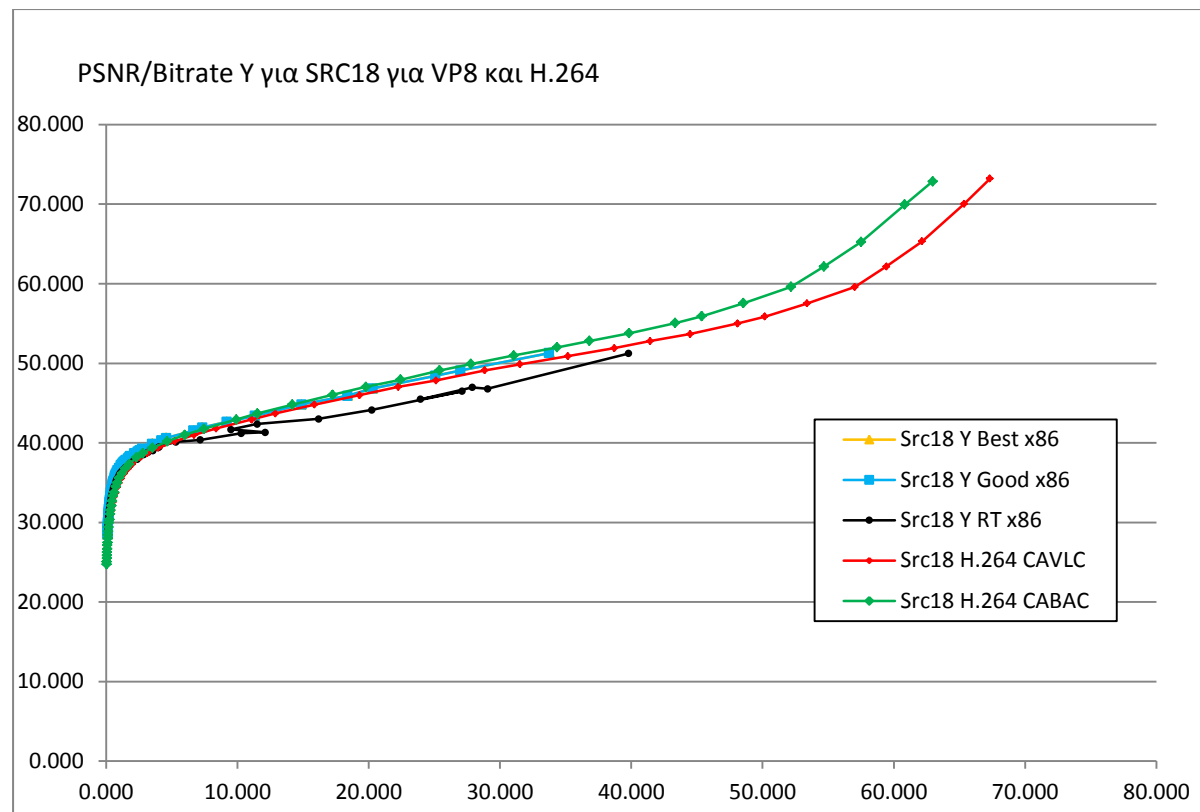
Σχήμα 22.7.15: PSNR/Bitrate Y για SRC15 για VP8 και H.264



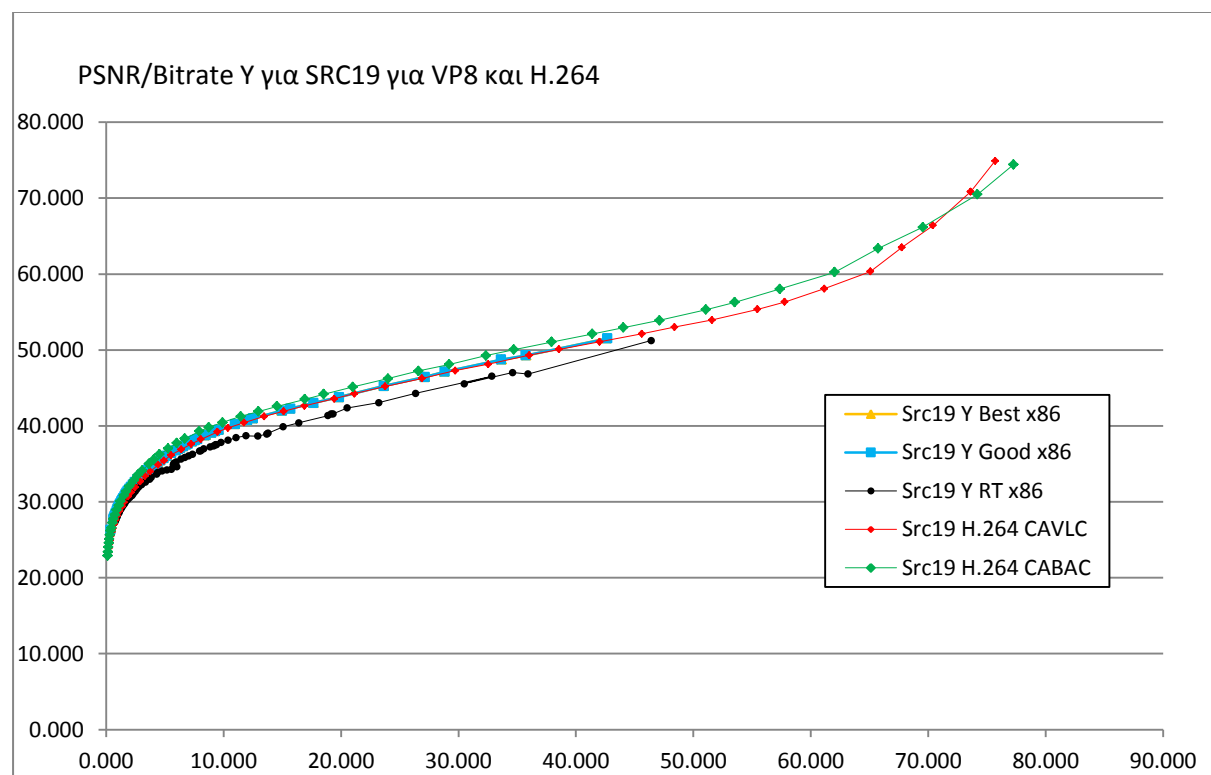
Σχήμα 22.7.16: PSNR/Bitrates Y για SRC16 για VP8 και H.264



Σχήμα 22.7.17: PSNR/Bitrates Y για SRC17 για VP8 και H.264

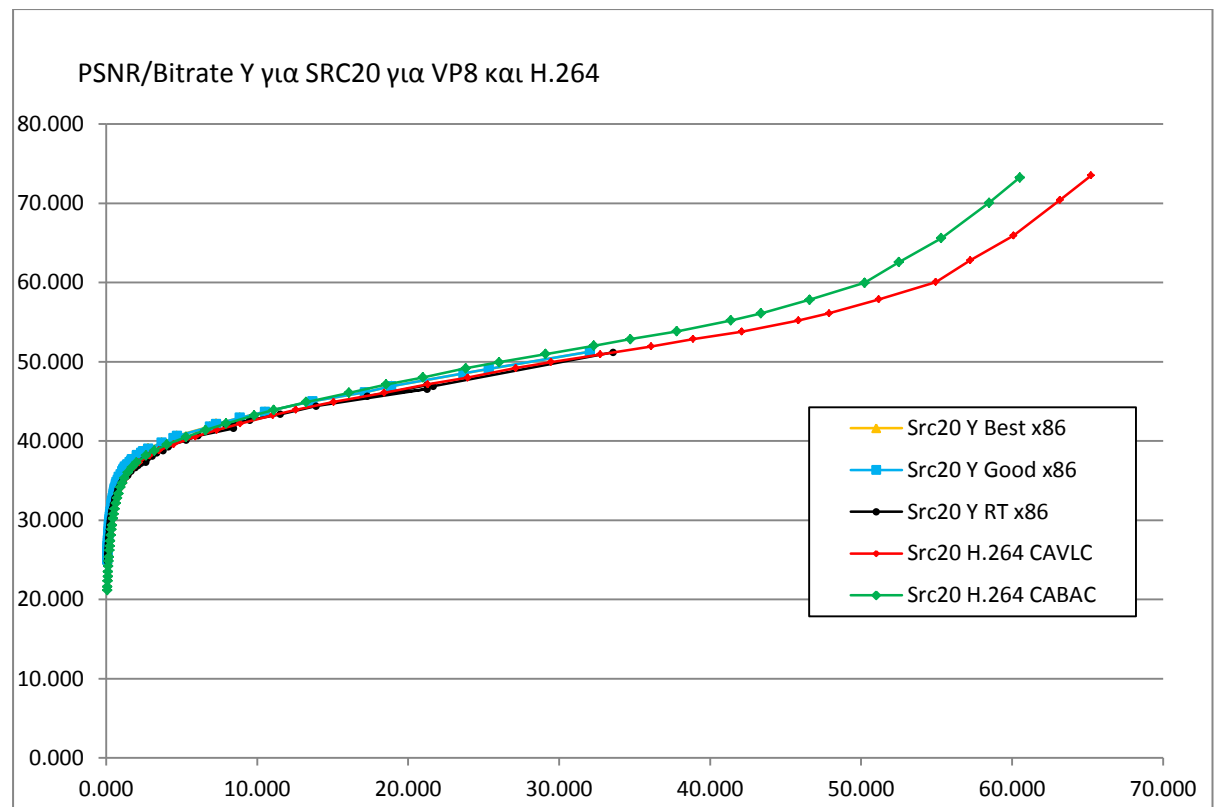


Σχήμα 22.7.18: PSNR/Birate Y για SRC18 για VP8 και H.264

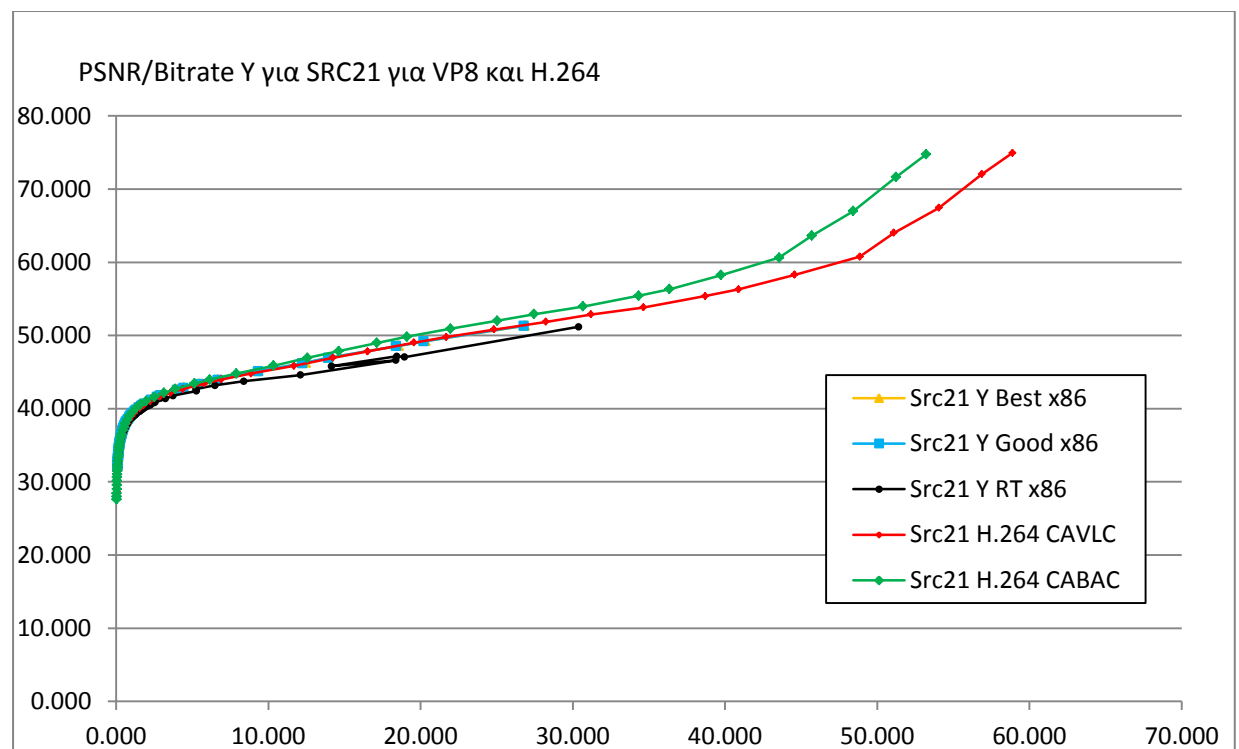


Σχήμα 22.7.19: PSNR/Birate Y για SRC19 για VP8 και H.264

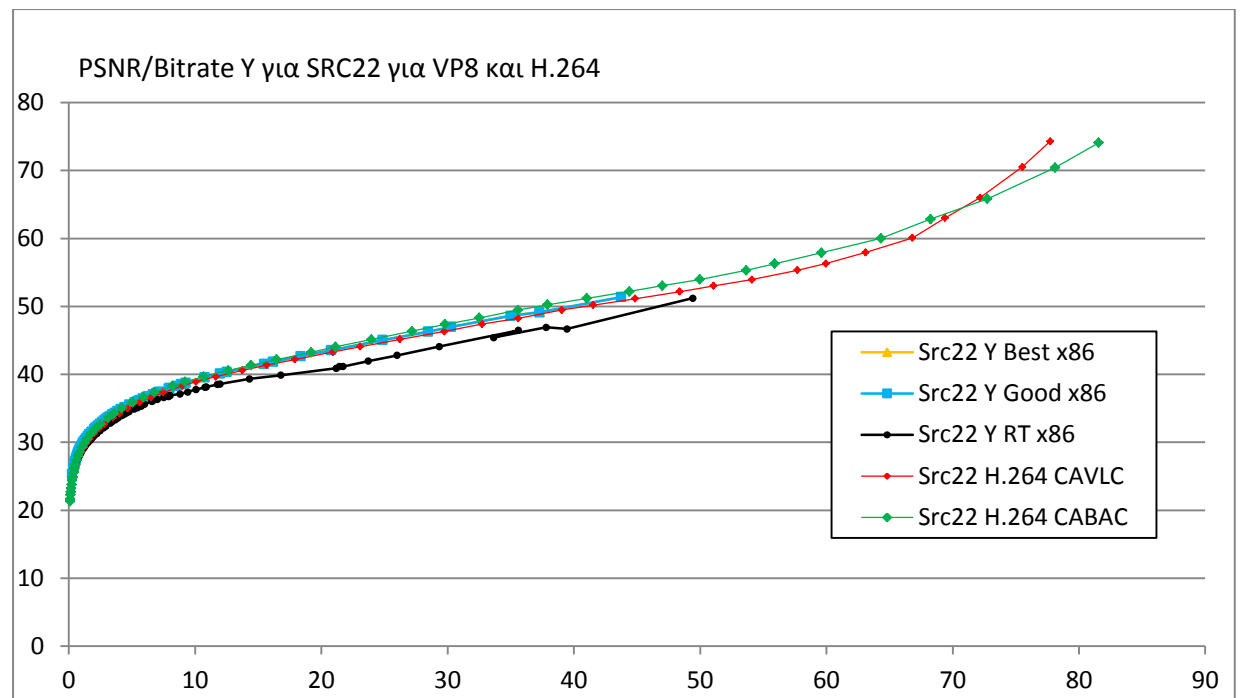




Σχήμα 22.7.20: PSNR/Bitrates Y για SRC20 για VP8 και H.264



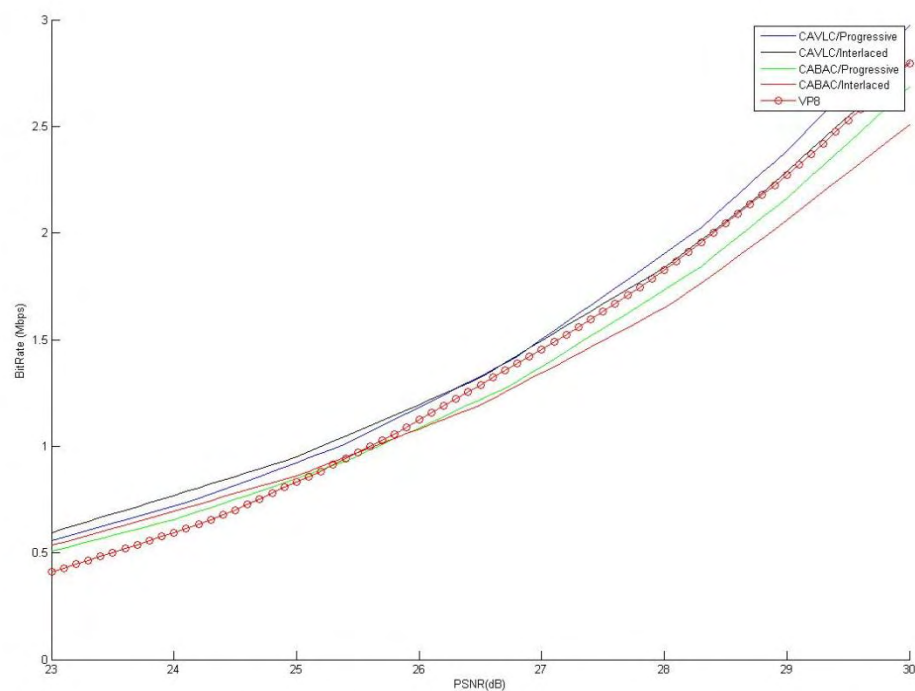
Σχήμα 22.7.21: PSNR/Bitrates Y για SRC21 για VP8 και H.264



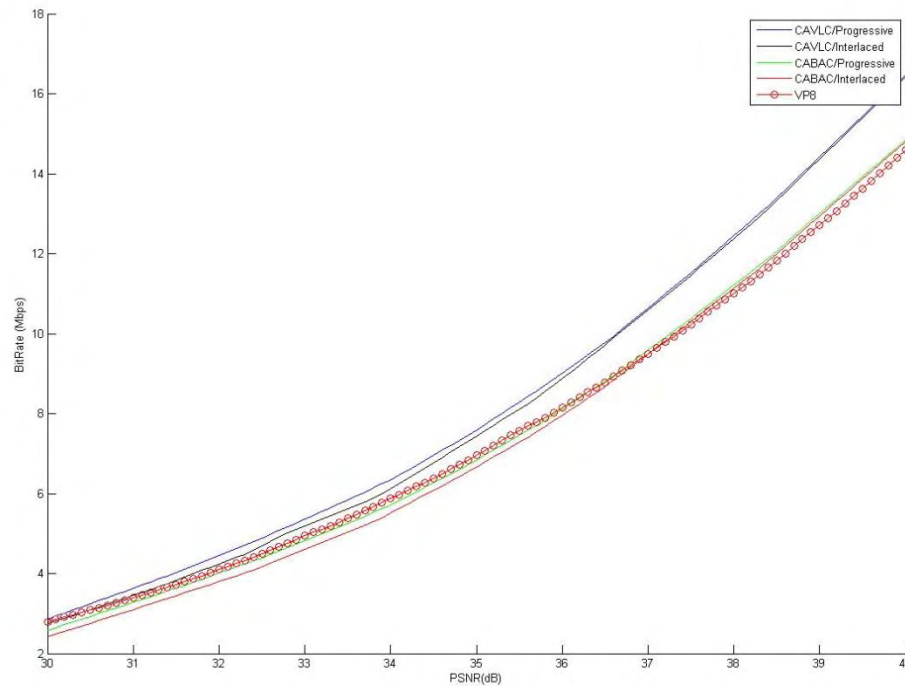
Σχήμα 22.7.22: PSNR/Bitrate Y για SRC22 για VP8 και H.264

Παρατηρούμε ότι ο VP8 εμφανίζει άνω όριο ποιότητας στα 52 dB σε σχέση με τα 75 dB που επιτυγχάνεται από το H.264.

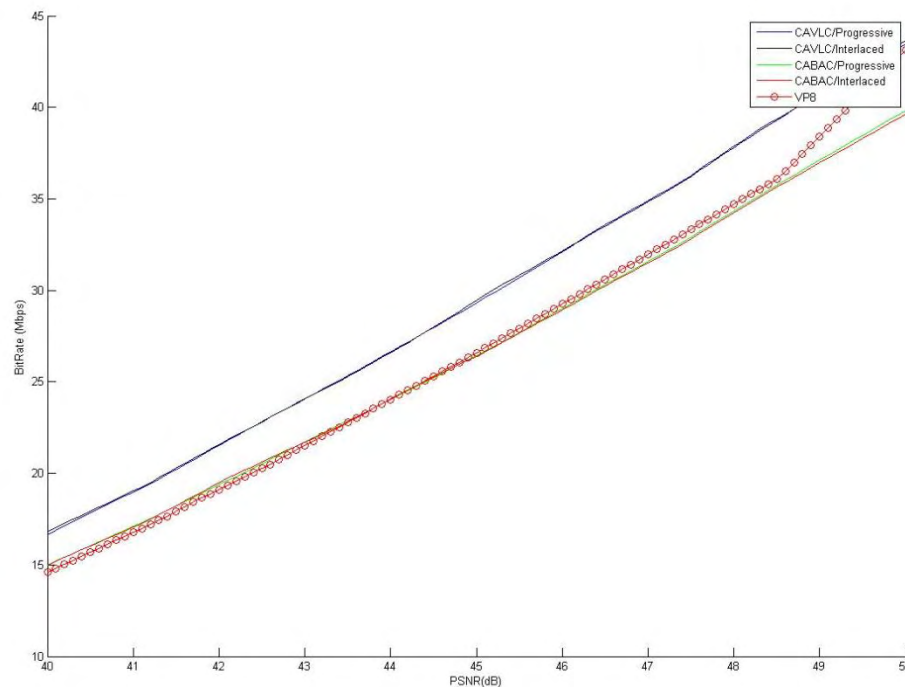
Επίσης κατά μέσο όρο σε επίπεδα bitrate μεγαλύτερα από 1Mbps υστερεί κατά 0,7 dB από τον h.264 όμως σε επίπεδα μικρότερα 1Mbps υπερτερεί. Για παράδειγμα, ας δούμε μια αναλυτική παρουσίαση του γραφήματος Bitrate/PSNR για το src15.



Σχήμα 22.8.2: Bitrate/PSNR για Y για SRC15 για VP8 και H.264 για εύρος 20 db έως 30 db



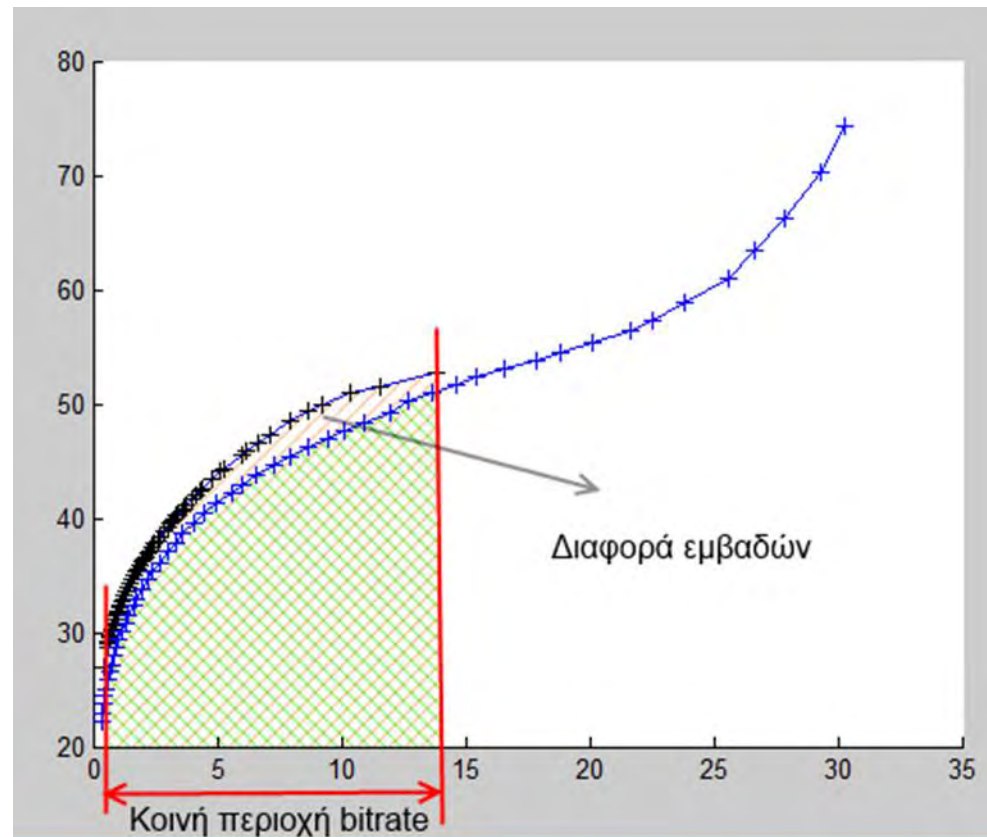
Σχήμα 22.8.2: Bitrate/PSNR για Y για SRC15 για VP8 και H.264 για εύρος 30 db έως 40 db



Σχήμα 22.8.3: Bitrate/PSNR για Y για SRC15 για VP8 και H.264 για εύρος 40 db έως 50 db

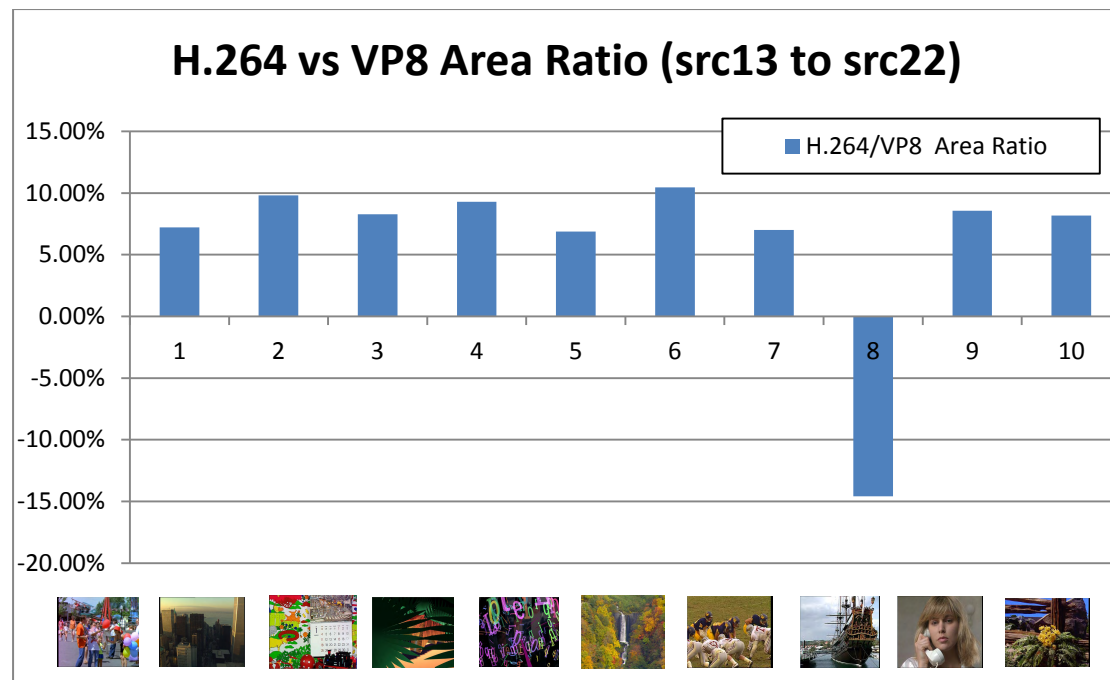
Επίσης για όλο το dataset έγινε σύγκριση για συγκεκριμένα διαστήματα ποιότητας PSNR. Το πρώτο διάστημα μεταξύ 50db και 40db (High quality) και το δεύτερο μεταξύ 40db και 30db (Low quality). Στο πρώτο διάστημα ο VP8 είναι ~3% χειρότερος από τον H.264, δηλαδή κατά μέσο όρο για όλα τα βίντεο του dataset ο VP8 απαιτεί 3% περισσότερο bitrate για το ίδιο PSNR. Αντίστοιχα στο δεύτερο διάστημα (40db με 30db ) παρατηρήσαμε ότι ο VP8 είναι ~7% καλύτερος από τον H.264, δηλαδή ο VP8 απαιτεί 7% λιγότερο bitrate για το ίδιο PSNR.

Επίσης υπολογίστηκε η διαφορά ποιότητας μεταξύ H.264 CABAC και VP8 Best για τις κοινές περιοχές bitrate για κάθε βίντεο. Για τον υπολογισμό της, πήραμε την διαφορά εμβαδών των γραφημάτων PSNR/Bitrate και για επικαλυπτόμενες περιοχές bitrate.



Σχήμα 22.9: Υπολογισμός διαφοράς εμβαδών

Και στην συνέχεια την διαφορά την εκφράσαμε ως ποσοστιαία μεταβολή με τον ακόλουθο τύπο  $((\text{εμβαδόν\_vp8} - \text{εμβαδόν\_h264}) / \text{εμβαδόν\_vp8}) * 100$ . Τα αποτελέσματα φαίνονται στο παρακάτω σχήμα



Σχήμα 22.10: Διαφορά ποιότητας μεταξύ H.264 CABAC και VP8 Best

Τέλος σε επίπεδο χρόνου, ο κωδικοποιητής VP8 είναι ταχύτερος, περίπου 25 φορές πιο γρήγορος από τον H.264 για παρόμοια ποιότητα video.

### 6.5.3 Αρχιτεκτονική 64 bit

Οι επεξεργαστές της AMD όσο και της INTEL στους προσωπικούς υπολογιστές υποστηρίζουν την αρχιτεκτονική 64-bit την οποία η AMD ονομάζει “AMD64” [14]. Τα πλεονεκτήματα της χρήσης της αρχιτεκτονική 64 bit είναι τα ακόλουθα:

- **Ακέραιοι 64-bit:** Όλοι οι καταχωρητές γενικής χρήσης επεκτείνονται από τα 32-bit στα 64-bit και όλες οι αριθμητικές και λογικές λειτουργίες και οι λειτουργίες μεταξύ καταχωρητών και μνήμης μπορούν να χρησιμοποιούν ακεραίους 64-bit.
- **Επιπλέον καταχωρητές:** Εκτός από την αύξηση του μεγέθους των καταχωρητών γενικής χρήσης, ο αριθμός των καταχωρητών γενικής χρήσης αυξάνεται από τους **8** (eax, ebx, ecx, edx, ebp, esp, esi, edi) του x86 σε **16** (rax, rbx, rcx, rdx, rbp, rsp, rsi, rdi, r8, r9, r10, r11, r12, r13, r14, r15).

- **Επιπλέον καταχωρητές XMM (SSE):** Ο αριθμός των καταχωρητών XMM 128-bit αυξάνεται επίσης από 8 σε 16 καθώς και το εύρος τους από 128-bit σε 256-bit [12] [13].
- **Μεγαλύτερος χώρος εικονικών διευθύνσεων:** Η αρχιτεκτονική AMD64 ορίζει έναν τύπο εικονικών διευθύνσεων 64-bit, τα 48 bit χαμηλότερης τάξης του οποίου χρησιμοποιούνται στις περισσότερες υλοποιήσεις

Στις μετρήσεις που έγιναν κατέληξα στα ακόλουθα συμπεράσματα :

- VP8: Το 64bit εκτελέσιμο δεν παρουσιάζει βελτιωμένη απόδοση σε σχέση με το 32bit. Χρειάζεται τροποποιήσεις για να εμφανίσει βελτιστοποίηση για 64bit. Χρήζει περεταίρω έρευνας.
- H.264: **1,8** φορές γρηγορότερη κωδικοποίηση όταν το εκτελέσιμο γίνεται compile για πλατφόρμα x64 χωρίς κάποια άλλη βελτιστοποίηση.
- H.265 : Παρόμοιο κέρδος με τον H.264 (~1.8x).

#### 6.5.4 Multithreading

Ο κώδικας αναφοράς του VP8 υποστηρίζει την χρήση νημάτων τόσο στον κωδικοποιητή όσο και στον αποκωδικοποιητή.

```
--threads=<arg> (or -t <arg>) (recommended value : number of real cores - 1)
```

Σχήμα 23: Παράμετρος ενεργοποίησης εκτέλεσης multithreading

Η παράμετρος `--threads` ορίζει τον αριθμό των νημάτων που θα χρησιμοποιηθούν κατά την διαδικασία της κωδικοποίησης. Ο VP8 υποστηρίζει έναν μηχανισμό όπου οι γραμμές των macro-blocks μπορούν να κωδικοποιηθούν ταυτόχρονα σε ξεχωριστά νήματα. Ωστόσο, το στάδιο του entropy encoding περιορίζεται στην χρήση ενός νήματος, εκτός αν χρησιμοποιηθεί μια δεύτερη παράμετρος, η `--token-parts`. Αξίζει να σημειωθεί ότι αν ο αριθμός των νημάτων οριστεί μεγαλύτερος του ενός >1 τότε το αποτέλεσμα διαδοχικών κωδικοποιήσεων με τις ίδιες παραμέτρους δεν θα είναι πάντα ακριβώς το ίδιο [18].

Αυτό επιβεβαιώθηκε και από την παρακάτω εκτέλεση



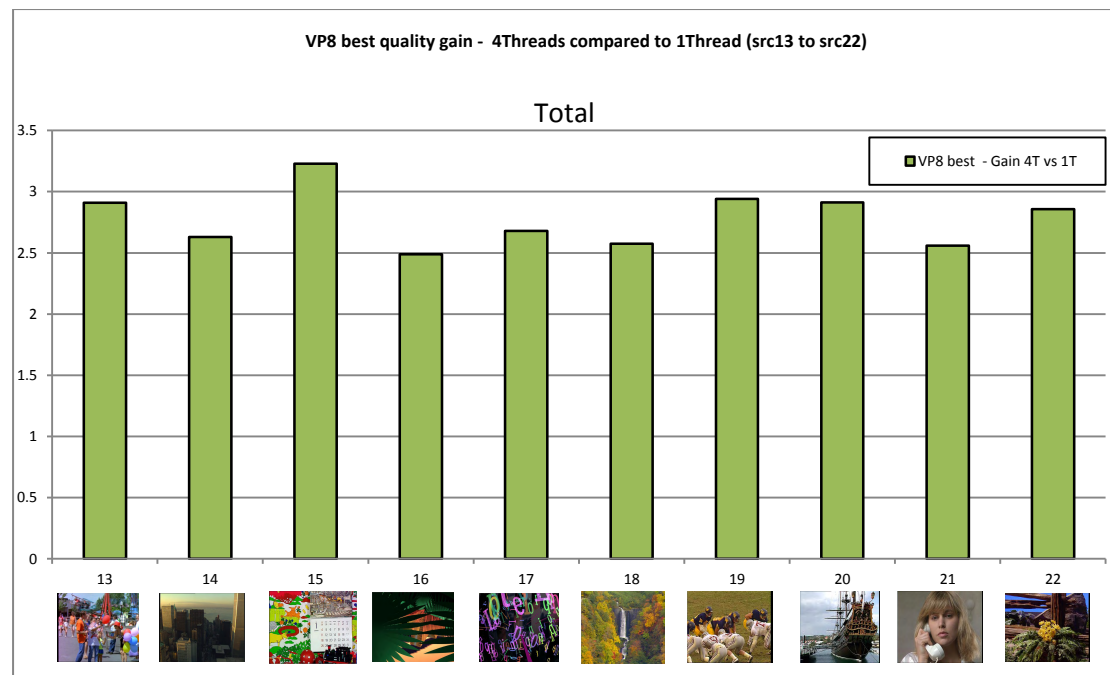
```

C:\Windows\system32\cmd.exe
D:\vpx-vp8-debug-src-x86-win32mt-vs9-v1.0.0\build\Win32\Release>vpxenc -t 4 -en
d-usage=vbr --best --psnr --min-q=32 --max-q=32 --limit=260 -w 720 -h 480 -o src
15_out_vpx_release_QP_%G.webm src15_ref_720x480_420.yuv
Pass 1/1 frame 260/260 5110857B 157257b/f 4717714b/s 158884 ms (1.64 fps)+IK
PSNR (Overall/Avg/Y/U/V) 33.602 33.603 32.738 35.994 36.280
D:\vpx-vp8-debug-src-x86-win32mt-vs9-v1.0.0\build\Win32\Release>vpxenc -t 4 -en
d-usage=vbr --best --psnr --min-q=32 --max-q=32 --limit=260 -w 720 -h 480 -o src
15_out_vpx_release_QP_%G.webm src15_ref_720x480_420.yuv
Pass 1/1 frame 260/260 5111897B 157289b/f 4718674b/s 141725 ms (1.83 fps)+IK
PSNR (Overall/Avg/Y/U/V) 33.603 33.604 32.739 35.994 36.281
D:\vpx-vp8-debug-src-x86-win32mt-vs9-v1.0.0\build\Win32\Release>vpxenc -t 4 -en
d-usage=vbr --best --psnr --min-q=32 --max-q=32 --limit=260 -w 720 -h 480 -o src
15_out_vpx_release_QP_%G.webm src15_ref_720x480_420.yuv
Pass 1/1 frame 260/260 5113088B 157325b/f 4719773b/s 151006 ms (1.72 fps)+IK
PSNR (Overall/Avg/Y/U/V) 33.601 33.602 32.738 35.994 36.282
D:\vpx-vp8-debug-src-x86-win32mt-vs9-v1.0.0\build\Win32\Release>vpxenc -t 4 -en
d-usage=vbr --best --psnr --min-q=32 --max-q=32 --limit=260 -w 720 -h 480 -o src
15_out_vpx_release_QP_%G.webm src15_ref_720x480_420.yuv
Pass 1/1 frame 260/260 5114150B 157358b/f 4720753b/s 154538 ms (1.68 fps)+IK
PSNR (Overall/Avg/Y/U/V) 33.602 33.604 32.739 35.992 36.281
D:\vpx-vp8-debug-src-x86-win32mt-vs9-v1.0.0\build\Win32\Release>vpxenc -t 4 -en
d-usage=vbr --best --psnr --min-q=32 --max-q=32 --limit=260 -w 720 -h 480 -o src
15_out_vpx_release_QP_%G.webm src15_ref_720x480_420.yuv
Pass 1/1 frame 260/260 5117770B 157469b/f 4724095b/s 143857 ms (1.81 fps)+IK
PSNR (Overall/Avg/Y/U/V) 33.603 33.604 32.739 35.991 36.289
D:\vpx-vp8-debug-src-x86-win32mt-vs9-v1.0.0\build\Win32\Release>vpxenc -t 4 -en
d-usage=vbr --best --psnr --min-q=32 --max-q=32 --limit=260 -w 720 -h 480 -o src
15_out_vpx_release_QP_%G.webm src15_ref_720x480_420.yuv
Pass 1/1 frame 260/260 5116540B 157432b/f 4722960b/s 143717 ms (1.81 fps)+IK
PSNR (Overall/Avg/Y/U/V) 33.604 33.605 32.741 35.998 36.279
D:\vpx-vp8-debug-src-x86-win32mt-vs9-v1.0.0\build\Win32\Release>

```

Σχήμα 24 : Διαφορετικά μεγέθη κωδικοποιημένου βίντεο κατά VP8 με ίδιες παραμέτρους με πολυνηματική εκτέλεση

Το κέρδος ταχύτητας όταν αξιοποιείται το multithreading και συγκεκριμένα χρησιμοποιώντας 4 πυρήνες αντί του 1 φαίνεται στο ακόλουθο διάγραμμα.



Σχήμα 25: Κέρδος ταχύτητας multithreading για τον VP8





## 7 Μελλοντικοί codec

### 7.1 H.265

Ο High Efficiency Video Coding (HEVC) είναι ο διάδοχος του H.264/MPEG-4 AVC codec . Βασίζεται στην ανάπτυξη του ISO/IEC 23008-2 MPEG-H Part 2 από το ISO/IEC Moving Picture Experts Group (MPEG) και του ITU-T H.HEVC από το the ITU-T Video Coding Experts Group (VCEG).

Οι δυο αυτές ομάδες εργασίας ενώθηκαν δημιουργώντας το MPEG & VCEG Joint Collaborative Team on Video Coding (JCT-VC), η πρώτη συνάντηση του οποίου έγινε τον Απρίλιο του 2010.

Οι πρωταρχικές απαιτήσεις που τέθηκαν ήταν:

- Μείωση του bit rate στο 50% για την ίδια υποκειμενική ποιότητα εικόνας σε σχέση με το High profile του H.264/MPEG-4 AVC
- Με υπολογιστική πολυπλοκότητα μεταξύ  $\frac{1}{2}$  και 3 φορές από εκείνη του High profile του H.264/MPEG-4 AVC.

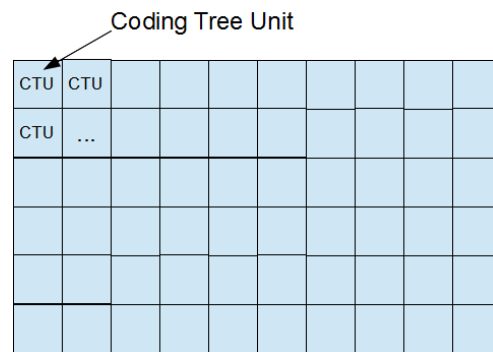
Το χρονοδιάγραμμα για την ολοκλήρωση του HEVC στάνταρ είναι το ακόλουθο :

- Φεβρουάριος 2012: Προσχέδιο Επιτροπής (ολοκλήρωση του προσχέδιου του στάνταρ)
- Ιούλιος 2012: Διεθνές Στάνταρ Προσχεδίου
- January 2013: Τελικό Διεθνές Στάνταρ Προσχεδίου Final Draft International Standard [19]

#### Prediction block size

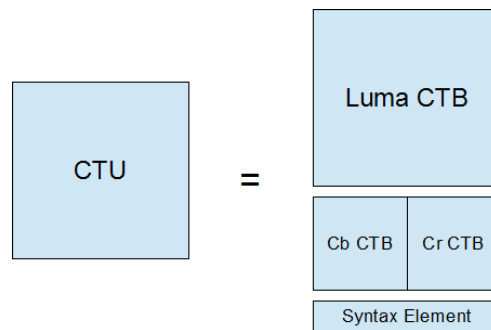
- Ο HEVC αντικαθιστά τα macroblocks, που χρησιμοποιούνται στον H.264 (μεγέθους 16x16), με ένα νέο σχήμα κωδικοποίησης που χρησιμοποιεί μεγαλύτερες δομές μπλοκ μεγέθους μέχρι 64x64 pixels και μπορούν να τμηματοποιήσουν καλύτερα την εικόνα σε δομές μεταβλητού μεγέθους.

- Ο HEVC αρχικά χωρίζει την εικόνα σε coding tree units (CTUs)



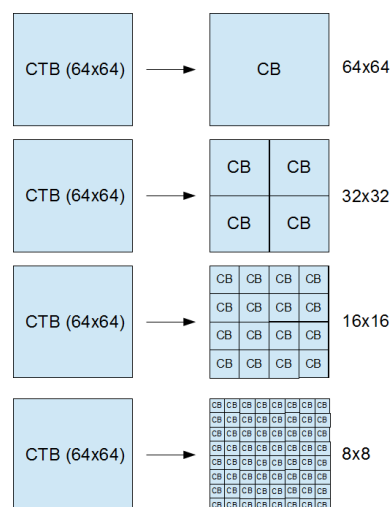
Σχήμα 26: Partitioning εικόνας σε CTU

- Τα CTUs χωρίζονται για κάθε συνιστώσα φωτεινότητας/χρωματικότητας σε coding tree blocks (CTBs)



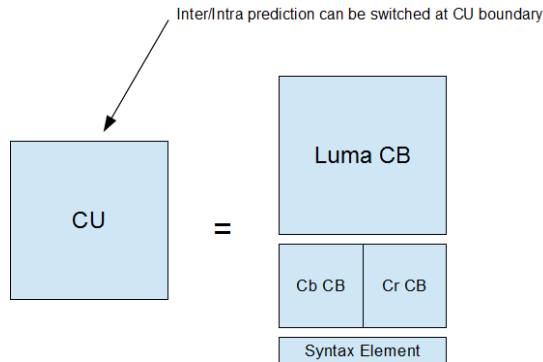
Σχήμα 27: CTU partitioning

- Τα CTB μπορούν να έχουν μέγεθος  $64 \times 64$ ,  $32 \times 32$ , or  $16 \times 16$ . Κάθε CTB μπορεί να διαιρεθεί ξεχωριστά σε πολλαπλά CBs (Coding Blocks).



Σχήμα 28: Μεγέθη CTB

- Ο τύπος πρόβλεψης κωδικοποιείται σε CU (Coding Unit). Το CU αποτελείται από τρία CBs (Y, Cb, and Cr)



Σχήμα 29: : CU partitioning

#### Αύξηση Internal bit depth

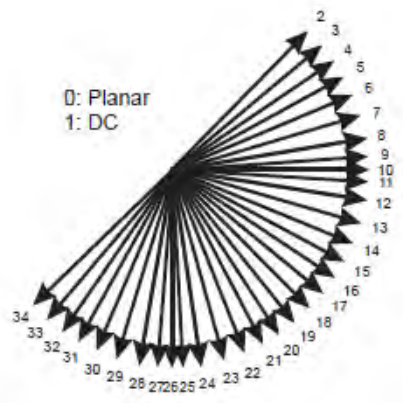
- Η Internal bit depth increase (IBDI) επιτρέπει να γίνεται η επεξεργασία των εικόνων εσωτερικά σε βάθος bit που είναι υψηλότερο από αυτό που θα κωδικοποιηθούν τελικά.
- Το IBDI μπορεί να γίνει σε βάθος μέχρι 14 bit και η επεξεργασία μπορεί να γίνει σε αυτό το βάθος μέχρι του σημείου που οι εικόνες τροφοδοτούν τα loop filters.

#### Entropy coding

- Ο HEVC χρησιμοποιεί context-adaptive binary arithmetic coding (CABAC) algorithm που είναι ουσιαστικά όμοιος με το CABAC στο H.264/MPEG-4 AVC
- Ο CABAC είναι η μόνη μέθοδος κωδικοποίησης εντροπίας που επιτρέπεται στον HEVC σε σχέση με τις δυο μεθόδους (CABAC και CAVLC) που επιτρέπονται στον H.264/MPEG-4 AVC.

#### Intra prediction

- Ο HEVC προσδιορίζει 33 ρυθμίσεις κατεύθυνσης για intra prediction σε σχέση με τις 8 ρυθμίσεις κατεύθυνσης που προσδιορίζονται από το H.264/MPEG-4 AVC. Επίσης προσδιορίζει planar και DC intra prediction modes.



Σχήμα 30: DC intra prediction modes

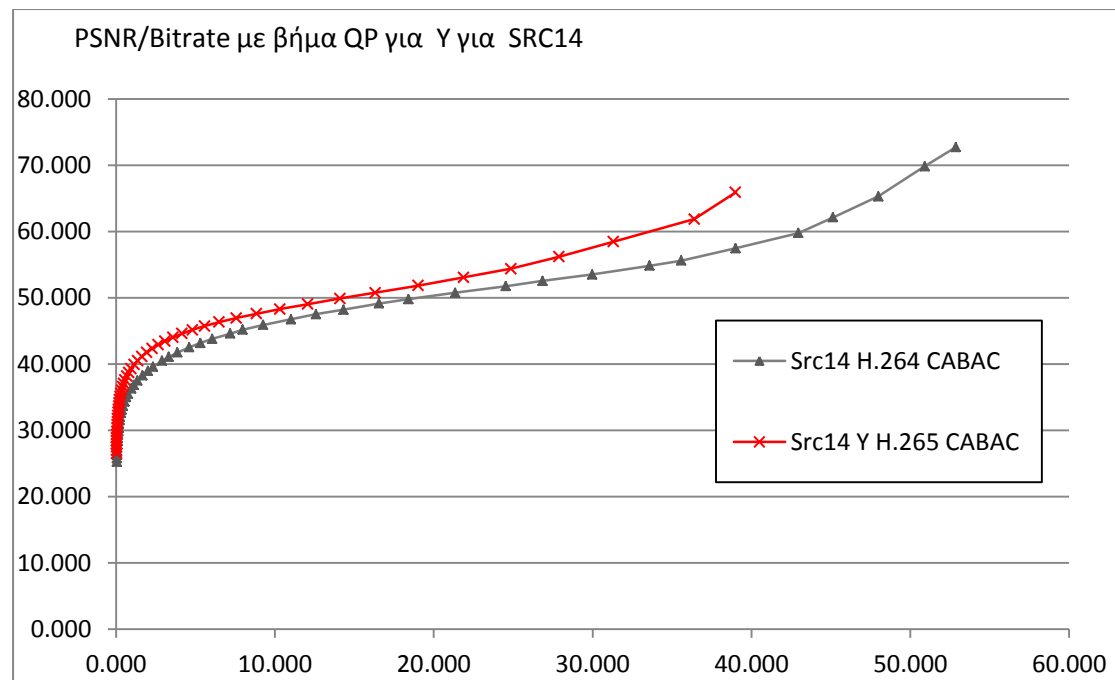
### Motion compensation

Χρησιμοποιείται ακρίβεια θέσης  $\frac{1}{4}$  δείγματος για τα διανύσματα κίνησης και φιλτράρισμα 7-tap ή 8-tap για παρεμβολή των θέσεων κλάσματος του δείγματος, σε σύγκριση με τα φίλτρα 6-tap θλών θέσεων μισού-δείγματος ακολουθημένο από διγραμμική παρεμβολή θέσεων  $\frac{1}{4}$  δείγματος στον H.264/MPEG-4 AVC

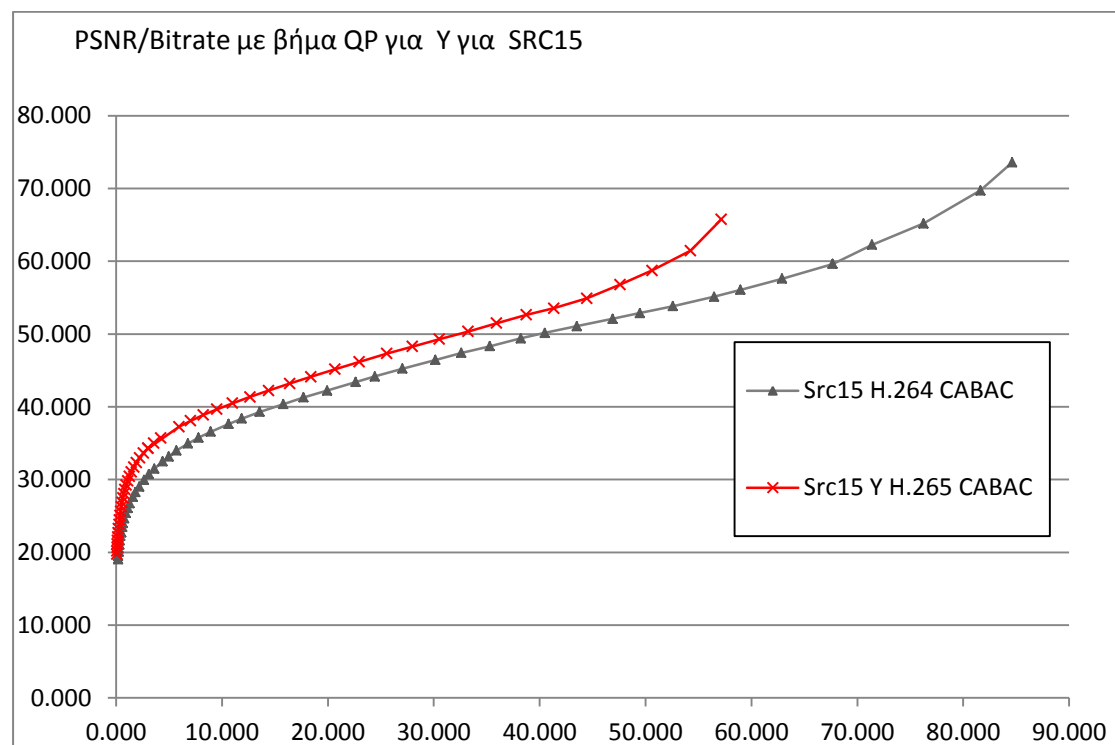
### In-loop deblocking filtering (DF)

Το deblocking filter, παρόμοιο με αυτό που χρησιμοποιείται στον H.264/MPEG-4 AVC, χρησιμοποιείται στο inter-picture prediction loop. Ωστόσο, ο σχεδιασμός του είναι απλοποιημένος όσον αφορά τις διαδικασίες φιλταρίσματος και λήψης αποφάσεων και είναι πιο φιλικό στην παράλληλη επεξεργασία.

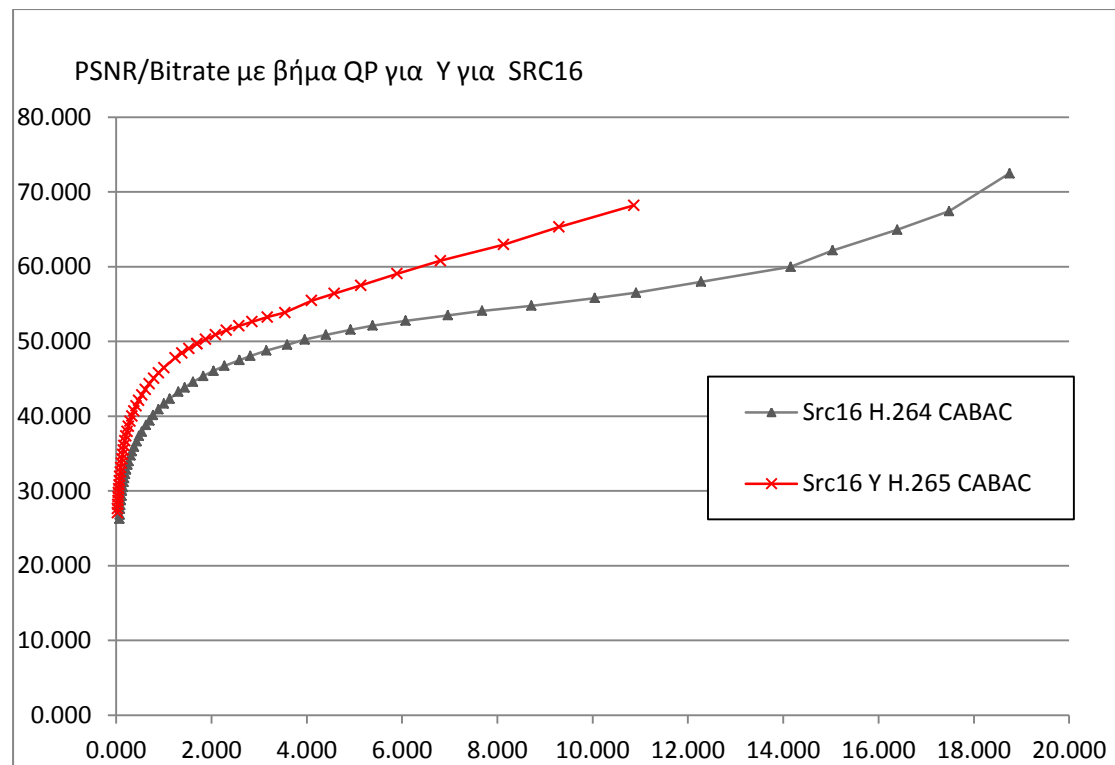
Στις μετρήσεις που έκανα, ο στόχος που τέθηκε σχετικά με την αποδοτικότητα του H.265, δηλαδή να πετύχει την ίδια ποιότητα με χρήση 50% του ρυθμού δεδομένων, σε κάποια βίντεο υπερκαλύπτεται και σε άλλα πλησιάζεται. Σε κάθε περίπτωση και δεδομένου ότι ο H.265 είναι νέος codec και ο κώδικας υπό ανάπτυξη το μέλλον του είναι άκρως ελπιδοφόρο. Ακολουθούν ορισμένα αποτελέσματα μετρήσεων.



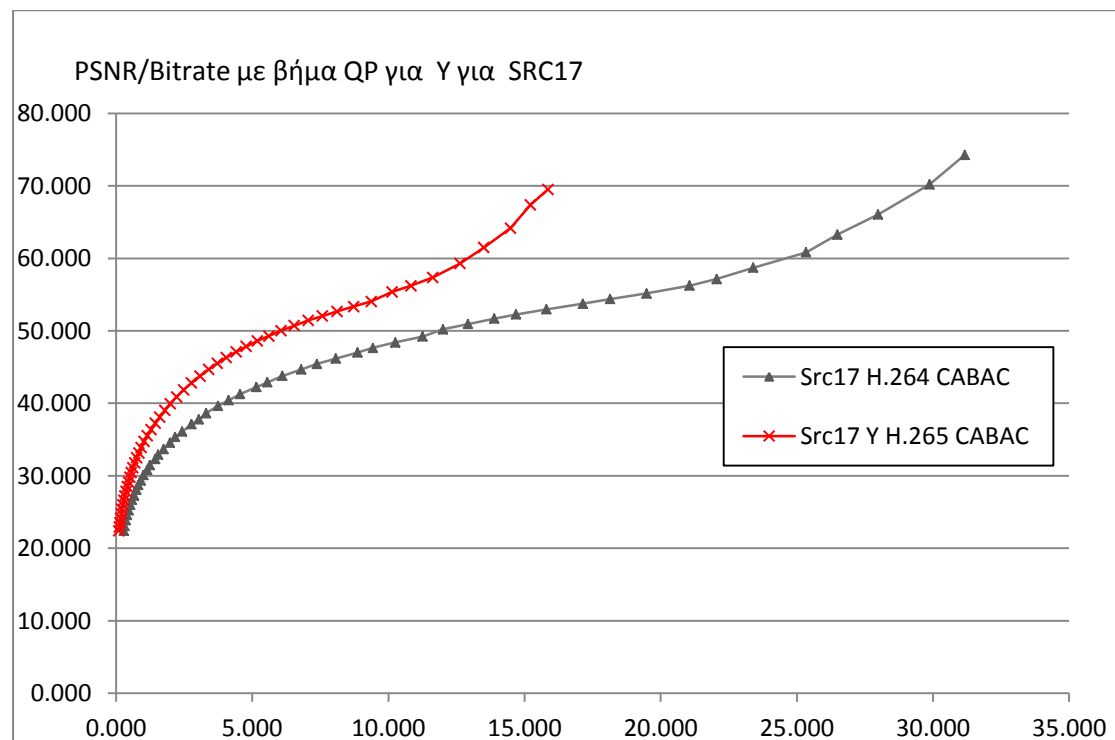
Σχήμα 31: Αποδοτικότητα H.265 σε σχέση με τον H.264 για src14.



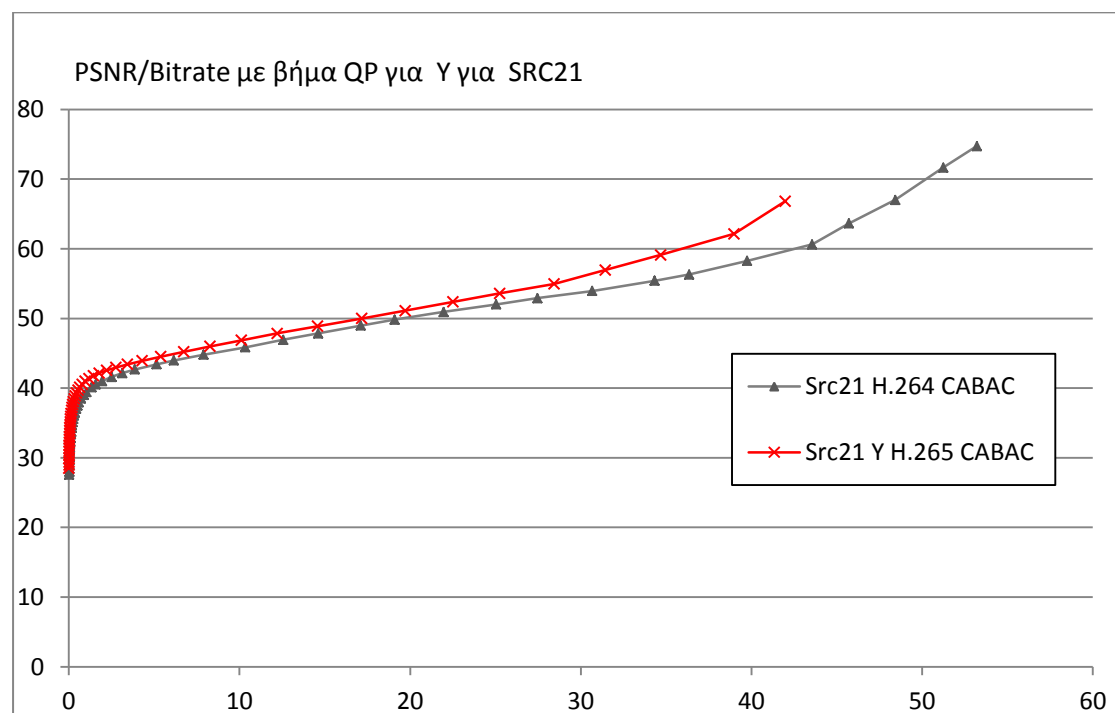
Σχήμα 32: Αποδοτικότητα H.265 σε σχέση με τον H.264 για src15.



Σχήμα 33: Αποδοτικότητα H.265 σε σχέση με τον H.264 για src16.



Σχήμα 34: Αποδοτικότητα H.265 σε σχέση με τον H.264 για src17. Επιβεβαιώνεται ο στόχος που τέθηκε ώστε να έχουμε την ίδια ποιότητα εικόνας με το μισό bitrate σε όλο το φάσμα του PSNR



Σχήμα 35: Αποδοτικότητα H.265 σε σχέση με τον H.264 για src21. Επιβεβαιώνεται ο στόχος που τέθηκε ώστε να έχουμε την ίδια ποιότητα εικόνας με το μισό bitrate

### H.264 vs H.265

- Ο H.265 επιτυγχάνει την ίδια ποιότητα video με την χρήση του μισού bitrate σε σχέση με τον H.264
- Ωστόσο αυτό γίνεται με το σχετικό κόστος σε χρόνο κωδικοποίησης +15%

## 7.2 VP9

Ο VP9 είναι ένα ανοικτό και χωρίς πνευματικά δικαιώματα πρότυπο που αναπτύσσεται από την Google και είναι ο διάδοχος του VP8. Ο VP9 είχε αρχικά ονομαστεί Next Gen Open Video (NGOV) καθώς και VP-Next.

Η ανάπτυξη του VP9 ξεκίνησε το τρίτο τρίμηνο του 2011. Ένας από τους στόχους του VP9 είναι να μειώσει το bit rate κατά 50% σε σχέση με το VP8 για την ίδια ποιότητα βίντεο.

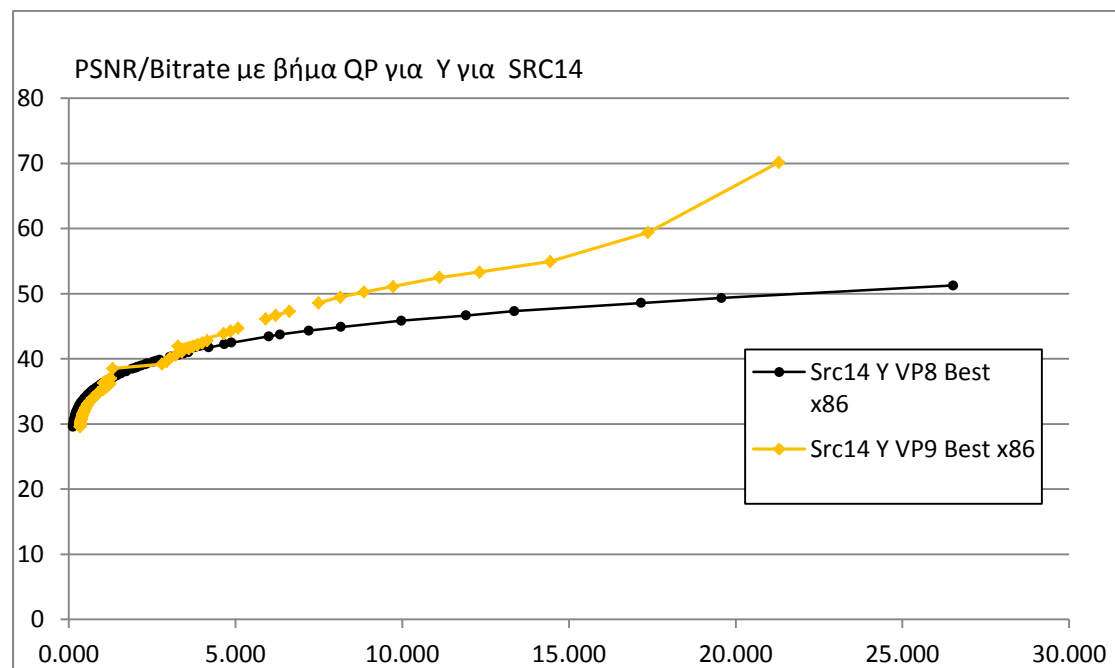


Άλλος στόχος είναι να βελτιωθεί σε τέτοιο σημείο όπου θα έχει καλύτερη αποδοτικότητα κωδικοποίησης από τον High Efficiency Video Coding.

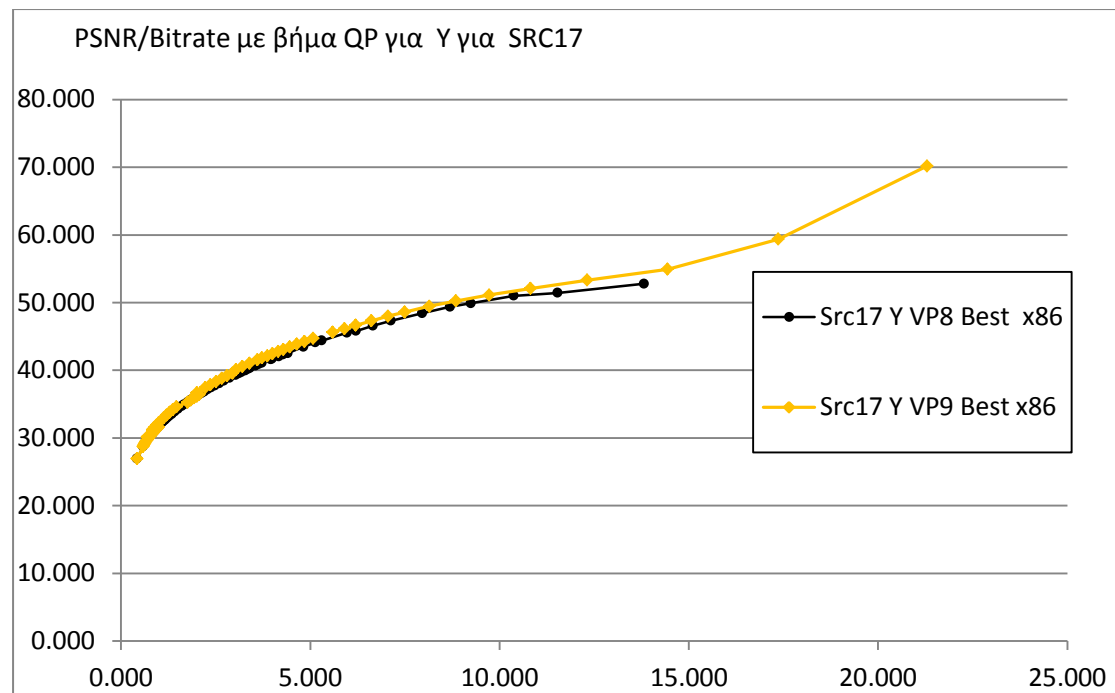
Έχει αρκετές σχεδιαστικές βελτιώσεις σε σχέση με τον VP8 μερικές από τις οποίες είναι η χρήση superblocks μεγέθους  $32 \times 32$  pixels (με πιθανή και την υποστήριξη  $64 \times 64$ ) και για τα οποία θα χρησιμοποιείται δομή κωδικοποίησης τετραδικού δένδρου.

Μέχρι και την ολοκλήρωση της διπλωματικής δεν είχε παρουσιαστεί ο πηγαίος κώδικας του VP9 αυτόνομα ως μέρος της βιβλιοθήκης libvpx. Για την δοκιμή του χρησιμοποιήθηκε το repository[17] location της developer edition του browser Google Chrome από το οποίο πήρα τον πηγαίο κώδικα και τον έκανα compile.

Στις μετρήσεις που έγιναν είχα τα εξής αποτελέσματα:



Σχήμα 36: Αποδοτικότητα VP9 σε σχέση με τον VP8 για src14.



Σχήμα 37: Αποδοτικότητα VP9 σε σχέση με τον VP8 για src17.

Παρατήρησα λοιπόν ότι σε μερικά βίντεο επιβεβαιώνεται ο στόχος που τέθηκε ώστε να έχουμε την ίδια ποιότητα εικόνας με το μισό bitrate σε σχέση με τον VP8 και σε άλλα όχι.



## 8 Βιβλιογραφία

- [1] Charles Poynton, “Digital Video and HD: Algorithms and Interfaces”, Morgan Kaufmann, 2002
- [2] Iain E. Richardson, “Video Codec Design: Developing Image and Video Compression Systems”, John Wiley & Sons, 17 Ιουνίου 2002
- [3] Chad Fogg. MPEG Video Compression Standard. Springer, 1996
- [4] S. Winkler, C. J. van den Branden Lambrecht, and M. Kunt (2001). "Vision and Video: Models and Applications". In Christian J. van den Branden Lambrecht. Vision models and applications to image and video processing. Springer. p. 209. ISBN 978-0-7923-7422-0
- [5] Iain E Richardson, “The H.264 Advanced Video Compression Standard”, John Wiley & Sons, 2010.
- [6] Marta Karczewicz and Ragip Kurceren, “The SP- and SI-Frames Design for H.264/AVC, IEEE Trans. on Circuit and System for Video Technology”, Vol.13, No. 7, Ιούλιος 2003
- [7] Bankoski, J. Google Inc., Mountain View, CA, USA Wilkins, P. ; Yaowu Xu “Technical overview of VP8, an open source video codec for the web”, Multimedia and Expo (ICME), 2011 IEEE International Conference on Date of Conference: 11-15 July 2011
- [8] Comparison of video resolutions [online], Εικόνα, Available: [http://en.wikipedia.org/wiki/File:Vector\\_Video\\_Standards2.svg](http://en.wikipedia.org/wiki/File:Vector_Video_Standards2.svg)
- [9] VQEG – Video Quality Experts Group , [online] Available: <http://www.its.bldrdoc.gov/vqeg/>
- [10] J. Bankoski, J. Koleszar, L. Quillio, J. Salonen, P. Wilkins, Y. Xu, “VP8 Data Format and Decoding Guide”, Google Inc., November 2011, [online], Available: <http://datatracker.ietf.org/doc/rfc6386/>
- [11] Thomas Maier, Dominik Hübner, Sven Pfleiderer, “The VP8 Video Codec Multimedia Codecs SS 2011”, [online], Available: <http://www.slideshare.net/pfleidi/the-vp8-video-codec>
- [12] 128-Bit and 256-Bit XOP and FMA4 Instructions [online], Available: [http://support.amd.com/us/Embedded\\_TechDocs/43479.pdf](http://support.amd.com/us/Embedded_TechDocs/43479.pdf)
- [13] Intel® Advanced Vector Extensions (Intel® AVX) [online], Available: <http://software.intel.com/en-us/avx>
- [14] AMD64 Architecture Programmer’s Manual Volume 2: System Programming [online], Available: [http://support.amd.com/us/Embedded\\_TechDocs/24593.pdf](http://support.amd.com/us/Embedded_TechDocs/24593.pdf)

- [15] VP8 Reference Implementation libvpx [online], Available:  
<https://code.google.com/p/webm/downloads/list>
- [16] H.264/AVC reference software [online], Available:  
<http://iphome.hhi.de/suehring/tml/download/>
- [17] libvpx 1.2.0 με υποστήριξη VP9 [online], Available:  
[https://src.chromium.org/chrome/trunk/deps/third\\_party/libvpx/source/libvpx/](https://src.chromium.org/chrome/trunk/deps/third_party/libvpx/source/libvpx/)
- [18] VP8 Encode Parameter Guide [online], Available:  
<http://www.webmproject.org/docs/encoder-parameters/>
- [19] International Organization For Standardization , MPEG HEVC – The next major milestone in MPEG video history is achieved,  
[http://mpeg.chiariglione.org/sites/default/files/files/meetings/docs/w13253\\_0.doc](http://mpeg.chiariglione.org/sites/default/files/files/meetings/docs/w13253_0.doc),  
103rd MPEG meeting, Geneva, Switzerland, 25 January 2013
- [20] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, Thomas Wiegand, «Overview of the High Efficiency Video Coding (HEVC) Standard» , Ieee Transactions On Circuits And Systems For Video Technology, Vol. 22, No. 12, December 2012
- [21] Ιωάννης Κατσαβουνίδης, «Τεχνικές Συμπίεσης Ήχου και Βίντεο», Διαφάνειες μαθήματος, Πανεπιστήμιο Θεσσαλίας